

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: M2612 – Elektrotechnika a informatika

Studijní obor: 3902T005 – Automatické řízení a inženýrská informatika

Sdílení dat mezi odloučeným pracovištěm a základnou

Data sharing between detached workplace and data base

Diplomová práce

Autor:	Oxana Shamanina
Vedoucí práce:	doc. Ing. Osvald Modrlák, CSc.
Konzultant:	Ing. Jan Pěnička

V Liberci 15.12. 2006

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

Anotace

Cílem diplomové práce je nalezení optimálního řešení ve sdílení dat na dálku. Existuje několik technologických možností takového sdílení dat. Problém nejrychlejší výměny informací mezi jednotlivými pracovišti je velmi aktuální; v současné době odborníci v různých oblastech hledají účinná řešení. Tato práce má za úkol prozkoumat všechna řešení a vybrat nejlepší.

Součástí této práce je zhotovená ukázková softwarová aplikace, jenž je navržena podle architektury klient-server. Klientem je moderní mobilní zařízení Personální Digitální Asistent (PDA), který se přes internet spojuje s webovou službou běžící na serveru, jenž je umístěn na vzdáleném pracovišti. Aplikace běžící na klientu má jednoduchou funkčnost. Zprostředkovává výměnu aktuálních dat mezi mobilním zařízením a databázovým serverem. Pro tvorbu této aplikace byly použity nejmodernější prostředky pro vývoj mobilních aplikací – platforma .NET.

Výsledkem této práce je moderní jednoduché řešení problému doplněné o názornou ukázkou jeho funkčnosti. V praxi by toto řešení mohlo být přínosem pro firmy, které potřebují rychle vyměňovat informace mezi jednotlivými pracovišti. Softwarová aplikace by mohla posloužit jako dobrý vzor pro vypracování dalších softwarových řešení pro potřeby různých firem v různých oborech.

Technologie přenosu dat na dálku má velkou budoucnost. Nepochybně se budou vyvíjet nové technologické možnosti pro bezdrátové sdílení dat. Snahou bylo vypracovat takové univerzální a jednoduché řešení, jenž by ho umožnilo při provedení drobných úprav využívat v mnoha oborech a přizpůsobit moderním trendům.

Abstract

The main aim of this diploma work is to find an optimal solution for wireless data sharing. The highest speed information exchange among individual workstations is a very actual problem. Many experts in different areas seek effective solutions to improve this nowadays. This work is to explore all solutions and then to select the best one.

A problem related to a technical point of a data transmission is not the only one solved within this diploma work, also a problem of a selection of a suitable transmission format is concerned. A data transmission security issue is not omitted as well.

There is a sample software application included in the work, which is designed on client-server architecture. A client is a smart mobile handler – Personal Digital Assistant (PDA), which through the Internet connects to a web service running on a server situated on a remote workstation. The application transmits actual data between a mobile handler and a database server. The most modern facilities were used to build this application – the platform .NET.

The result of this work is a smart simple problem solution, which is completed by a demonstration of its functionality. This solution could bring a benefit to companies that need a prompt information exchange among workstations.

Software solutions developed within the frame of this work is suitable for a building industry. It enables site managers both to update a company database and to reach the latest data on a server immediately.

Komunikace, sdílení dat, webové služby, remoting, PDA

Obsah

Prohlášení.....	3
Anotace	4
Abstrakt.....	4
Obsah	6
Seznam nejdůležitějších výrazů a zkratk.....	7
Úvod.....	8
1 Seznámení s problematikou a současnými řešeními	9
1.1 Využití mobilních technologií v moderním životě	9
1.1.1 Mobilní zařízení	9
1.1.2 Typy a platformy mobilních zařízení.....	14
1.2 Komunikace mezi vzdálenými zařízeními.....	16
1.2.1 Architektura bezdrátových sítí.....	16
1.2.2 Možností sdílení, techniky přenosu dat.....	17
1.2.3 Synchronizace dat	20
1.2.4 Bezdrátové připojení PDA k sítí	20
2 Technologické možnosti.....	23
2.1 Terminologie.....	23
2.2 Platforma pro vývoj	23
2.3 Stručná historie vývoje aplikací pro mobilní zařízení	26
2.4 Vývoj aplikací pro mobilní zařízení na platformě .NET	26
3 Definování toku dat v obou směrech komunikace a datového formátu	28
3.1 Možnosti komunikace mezi mobilním zařízením a databázovým serverem	28
3.1.1 .NET Remoting	28
3.1.2 Webové služby.....	32
3.1.3 Další způsoby připojení k serverové databázi.....	34
3.2 Způsoby synchronizace dat.....	37
3.3 Ukázková aplikace. Specifikace. Popis řešení.....	38
4 Použití IT technologií v oblasti stavebnictví	43
4.1 Test mobilních zařízení (PDA) na stavbě.....	43
4.2 Komunikace ve stavebnictví.....	44
4.3 Software ve stavebnictví.....	44
5 Praktická aplikace v podmínkách stavební firmy – stavbyvedoucí a sídlo firmy.....	47
Závěr	52
Seznam použité literatury.....	53
Příloha A – Specifikace IS RestSoft pro libovolné restaurační zařízení.....	54
Příloha B – Specifikace IS BuildingSoft pro stavbyvedoucího	69

Seznam nejdůležitějších výrazů a zkratek

.NET	platforma Microsoft .NET
.NET Remoting	soubor technik a služeb pro komunikaci aplikací na dálku přes TCP/IP nebo http
Bluetooth	komunikační technologie, která umožňuje bezdrátovou komunikací mezi zařízeními na krátkou vzdálenost
EDGE	Enhanced Data for GSM/GPRS Evolution
GPRS	General Packet Radio Service (Obecná paketová rádiová služba)
HSCSD	High Speed Circuit Switched Data
HTTP	HyperText Transfer Protocol
HTTPS	Hypertext Transfer Protocol over SSL
IIS	Internet Information Server
PDA	Personální Digitální Asistent
UMTS	Universal Mobile Telecommunication Systém
UDDI	Universal Description, Discovery and Integration
VPN	Virtual Private Network (virtuální privátní síť)
WAE	Wireless Application Environment
Webová služba	řešení pro komunikaci a výměnu informací aplikacemi přes Internet
Wi-Fi	wireless fidelity
WLAN	Wireless Local Area Network (bezdrátová místní síť)
XML	Extensible Markup Language

Úvod

Podnětem pro zadání této diplomové práce byla potřeba nalézt optimální řešení pro sdílení dat na dálku. V mnoha případech je potřeba, aby lidé byli mobilní bez omezení rozsahu komunikace. Komunikace s okolním prostředím je podstatná a musí být kvalitní. Množství a správné načasování výměny informací může buď bránit nebo naopak podporovat dosažení dobrých výsledků práce. S rozvojem mobilních telefonů a bezdrátové komunikace existuje více a více možností ke zefektivnění komunikace. Výběr způsobu komunikace na dálku je rozmanitý: mobilní telefony, hlasová VPN (Virtual Private Network), WLAN – bezdrátová místní síť, osobní mobilní rádio – vysílačky, moderní mobilní vysílačky (Push To Talk), satelitní technologie a další.

Tato práce se zaměřuje na nalezení mobilního řešení pro komunikaci v oblasti stavebnictví. Jedním z nejdůležitějších úkolů je vypracování jednoduché softwarové aplikace, která by umožňovala rychlé sdílení informací mezi odloučeným pracovištěm–stavbou a základnou–serverem v kanceláři stavební firmy. Představa je taková, že by každý stavbyvedoucí, zodpovědný za průběh konkrétní stavby, měl u sebe malé mobilní „zařízení do kapsy“ (tato softwarová aplikace je navržena pro zařízení PDA – Personální Digitální Asistent), na kterém je instalován nenáročný program. Tento program umožní zadávání nejdůležitějších informací do formulářů a následné bezdrátové spojení a odeslání dat na databázový server, který se nachází v kanceláři firmy a také obdržení nejaktuálnějších dat ze serveru přímo do místa působení.

Vzhledem ke specifikaci platformy Pocket PC (platforma pro zařízení PDA), která je určena k uchovávání dat v databázích přímo na klientu, tak nemusíme data odesílat a ukládat na vzdálený databázový server okamžitě (on-line synchronizace). Můžeme využít také možností offline synchronizace, kdy data jsou odesílána na server až po připojení zařízení k síti. Pro oba tyto účely můžeme využít výhod poskytovaných webovými službami, jenž běží na serveru a přístup k nim je umožněn prostřednictvím internetu – využívají se pouze povolené porty 80 (HTTP) a 443 (HTTPS), takže aplikace mezi sebou mohou nerušeně komunikovat. Druhá technologie přenosu dat na dálku se jmenuje .NET Remoting a je o něco propracovanější. V této práci se budeme zabývat oběma technologiemi. Popíšeme jejich vlastnosti a přednosti a ukážeme jak fungují v praxi .

K naprogramování softwarové aplikace pro bezdrátovou komunikaci jsou použity nejmodernější technologie. Program je vytvořen na platformě .NET od firmy Microsoft. Rámec .NET umožňuje efektivně vytvářet aplikace určené pro mobilní zařízení.

Idea této práce je taková, že je potřeba nalézt univerzální řešení pro sdílení dat na dálku, jenž je lehce implementovatelné, a v budoucnu jej bude možné použít nejen ve stavebnictví ale také v jakékoli jiné oblasti.

1 Seznámení s problematikou a současnými řešeními

Problematika přenosu dat mezi vzdálenými aplikacemi je velmi aktuální. Moderní styl života vyžaduje využití moderních technologií ve všech oblastech. Sdílení informace je velmi důležitou součástí našeho všedního a pracovního života. Lidé si brzy zvykli na rychlý přenos libovolných dat přes internet a v návaznosti na to vzrostly požadavky na výrobu mobilních aplikací pro přenos důležitých dat mezi vzdálenými oblastmi. Pracovníci potřebují mít trvalý a rychlý přístup k aktuálním datům odkudkoli: ze své kanceláře, ze vzdáleného pracoviště, z domova, z dovolené. I právě proto vznikala tato práce. Dále se podíváme na různé možnosti komunikace mezi mobilními zařízeními, způsoby sdílení dat a zprovoznění synchronizace informací.

1.1 Využití mobilních technologií v moderním životě

V poslední době čím dál větší popularitu získávají přenosné počítače třídy Handheld a Pocket PC. Tato skutečnost je podmíněna rozvojem internetu a mobilní komunikace. O osudu a úspěchu velkých obchodních transakcí často rozhodují doslova minuty. Každý se potřebuje operativně dostat ke své elektronické poště, případně pružně reagovat na různé podněty a informace. Někdy je zapotřebí vykonávat dálkový dohled nad nějakým technologickým zařízením, které pracuje v bezobslužném režimu. Na první pohled by se mohlo zdát, že zařízení této třídy budou jen „moderní hračkou“ v rukou manažerů, případně jim pomohou organizovat to, s čím každý den urputně bojují – čas; případně se na přístrojích této třídy dají hrát zajímavé kvalitní hry anebo číst knihy pomocí aplikace MSReader. Praxe však ukázala, že s vývojem komunikačních a hardwarových možností těchto zařízení, kdy jejich procesory jsou taktovány na stovkách MHz a paměťová kapacita je minimálně 64 MB, se podstatně rozšiřuje oblast jejich použití.

1.1.1 Mobilní zařízení

Co je vlastně mobilní zařízení? Většina lidí si pod pojmem „mobilní“ představuje něco, co je schopno komunikovat se sebe podobným nebo nějakým jiným zařízením na dálku. A také jako přístroj, který může být napájený z akumulátoru. Mnoho lidí by pravděpodobně požadovalo, aby takové zařízení bylo kompaktních rozměrů a umělo pracovat v rozmanitých prostředích. Podívejme se na různé druhy mobilních zařízení, způsoby jejich komunikace s okolním světem a možnosti využití v praxi.

Mobilní telefony

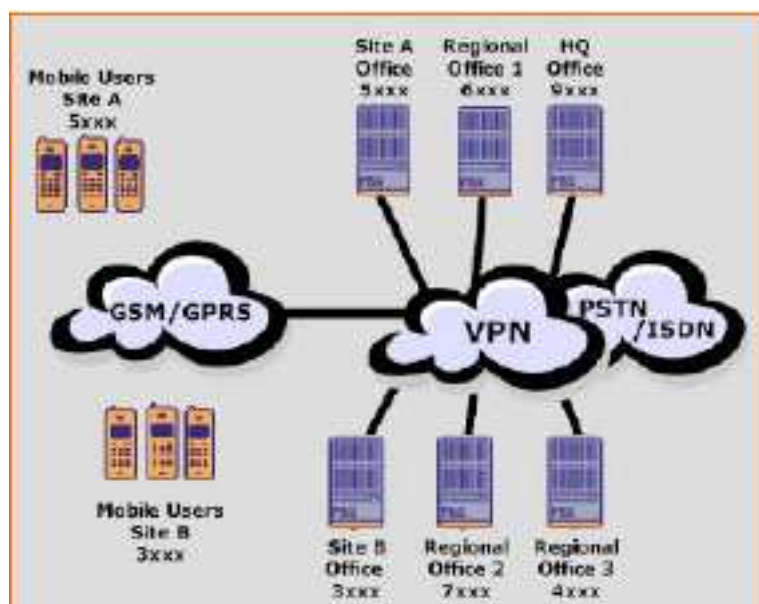
Mobilní telefon je nejrozšířenějším mobilním zařízením, má ho snad každý z nás pořád s sebou. Ale většina lidí zřejmě ani netuší, jaké možnosti nám nabízí moderní mobilní telefony.

Již dnes můžeme psát vlastní software pro tato zařízení, využívat je jako datovou základnu pro své aplikace nebo jako prostředek pro přenos důležitých pracovních dat na databázový server. Existuje mnoho způsobů zavedení této technologie jako komunikačního média. Nejjednodušší a nejvíce používaná je hlasová komunikace.



Obr. 1-1: Mobilní telefon je využíván všude, zdroj: The appropriate use of I.T. on a construction site

Alternativním řešením pro využití komunikace mobilních telefonů je provozovat **hlasovou VPN (Virtual Private Network)**. V takovém případě je každý mobilní telefon registrován u servisního poskytovatele a při volání pozná, že může použít hlasovou VPN a hovor je účtován centrálně dle dohodnutého paušálu.



Obr. 1-2: Schéma komunikace pomocí hlasové VPN, zdroj: The appropriate use of I.T. on a construction site

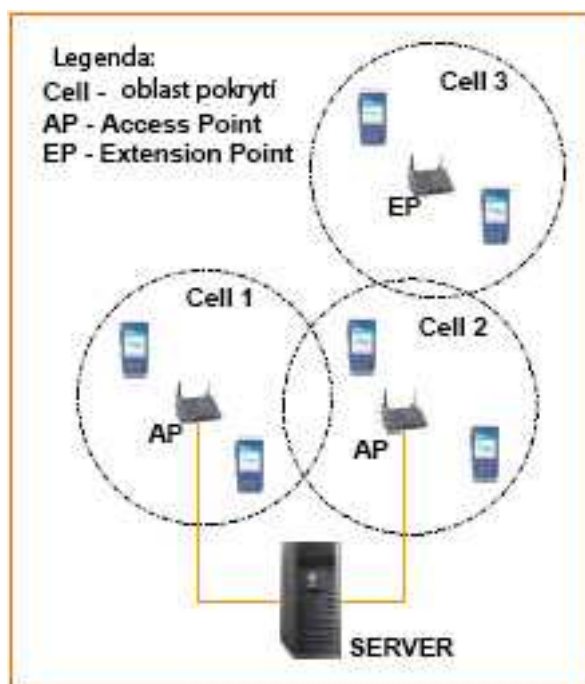
WLAN – bezdrátová místní síť

Bezdrátová LAN je normovaná na IEEE802.11b, IEEE802.11b a IEEE802.11g standardech. Novější technologií je IEEE802.11h

Technologie	Frekvenční pásmo/rychlost	Rychlost reálná	Dostupnost	Vzdálenost
IEEE802.11a	5GHz/54Mbps	25Mbps	US,UK, ne ČR	30m uvnitř- 300m venku
IEEE802.11b	2.4GHz/11Mbps	5Mbps	Dnes	30m uvnitř- 300m venku
IEEE802.11g	2,4GHz/54Mbps		2003- Dnes	Prudce klesá se vzdálenosti

Tab. 1-3: Přehled standardů bezdrátové místní sítě WLAN, zdroj: <http://www.lupa.cz>

Technologie WLAN využívá několika přenosových médií, nepoužívanější jsou mikrovlny. Wi-Fi technologie může být zavedena několika způsoby. První a nejjednodušší způsob je ad-hoc síť, kdy mobilní počítače se zabudovanou wi-fi kartou mohou komunikovat navzájem. Tato technologie umožňuje přenos informace mezi počítači, ale není povolen přístup k pevné síti. Nejpopulárnější způsob zapojení wi-fi je připojení mobilních uživatelů k hlavnímu serveru.



Obr. 1-4: Schéma komunikace pomocí WLAN, zdroj: <http://www.lupa.cz>

Osobní mobilní rádio – vysílačky

Používá se například ve stavebnictví posledních 15let. Umožňuje pouze hlasový přenos.



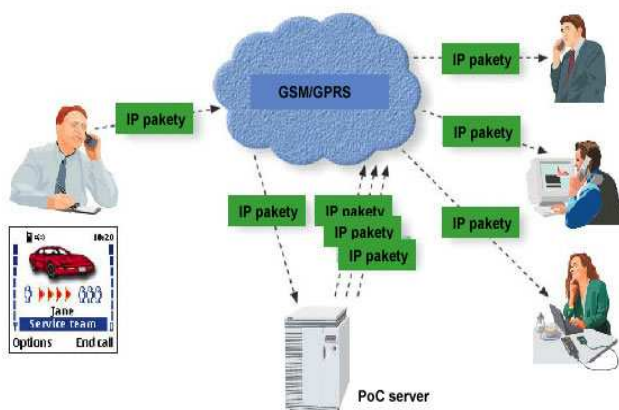
Obr. 1-5: Osobní mobilní rádio

Moderní mobilní vysílačky (Push To Talk)

Tato unikátní technologie rozšiřuje funkce mobilního telefonu o další hlasovou formu komunikace, která byla doposud doménou vysílaček.

Technologie Push to Talk umožňuje spojení s jedním nebo více účastníky najednou prostřednictvím GSM, respektive GPRS. Volající si vybere ze seznamu kontaktů jednotlivce nebo skupinu a stlačením speciálního tlačítka na mobilním přístroji může ihned po navázání spojení signalizovaného pípnutím začít mluvit. Spojení je téměř okamžité, bez vytáčení čísla a čekání na spojení hovoru.

Jako přenosová technologie pro Push to Talk v mobilní síti GSM slouží nejčastěji paketové spojení GPRS. V síti je kromě koncových zařízení nutná ještě přítomnost jednoho hlavního prvku, který celou komunikaci řídí. Tímto řídicím prvkem je Push to Talk application server, který se nejčastěji označuje jako PoC server. Základní princip systému je na následujícím obrázku.



Obr. 1-6: Princip komunikace Push To Talk, zdroj: <http://www.lupa.cz>

Satelitní technologie

Pomocí speciálních přístrojů je informace směřována na satelit a odtud je vysílána příjemci. Hlavní výhodou je pokrytí celé planety. Satelitní technologie umožňují jak hlasový, tak datový přenos.

Podívali jsme se na zajímavé možnosti, které se nám jako zákazníkům nabízejí pro uskutečnění komunikace a sdílení dat na dálku. Většina z nich však umožňuje pouze hlasovou komunikaci a pro naše účely je nepoužitelná. Proto z výše představených zařízení vybereme pouze mobilní telefon a podíváme se podrobněji na další druh mobilního zařízení PDA – Personální digitální asistent.

PDA

PDA jsou tzv. počítače do kapsy, které v poslední době vytlačují dosud používané papírové i elektronické diáře a organizéry. PDA toho ale umí daleko více než elektronický diář. Hlavní výhodou tohoto zařízení je jeho funkčnost. Pro nás nejdůležitější funkcí tohoto přístroje je operační systém, na jehož bázi můžeme psát vlastní aplikace. Otázka kompaktních rozměrů také není zanedbatelná. K dispozici jsou cenově rozdílné modely s černobílým i barevným displejem. Lze dokoupit rovněž množství různých doplňků – klávesnice, paměťové karty, modemy a podobně. Tento typ mobilního přístroje v sobě většinou má infračervený port (IrDA), bluetooth nebo Wi-fi kartu pro připojení kapesního počítače k osobnímu počítači nebo k internetu. PDA zařízení může sloužit i jako mobilní telefon. Pro ukázkové aplikace této diplomové práce bylo vybráno právě toto zařízení, hlavně pro jeho jednoduchost, kompaktnost a funkčnost. Společnost Microsoft se už delší dobu zabývá podporou vývoje aplikací pro kapesní počítače a jiné mobilní zařízení. Existují také různé druhy emulátorů, které umožňují vývojářům vytvářet software pro kompaktní počítače, aniž by vlastnili samotné zařízení.



Obr. 1-7: PDA(Personál Digital Asistent), zdroj Microsoft

1.1.2 Typy a platformy mobilních zařízení

Aby jsme měli přehled o jednotlivých třídách mobilních klientů nejprve uvedeme stručný přehled platforem z dílny Microsoft.



Obr. 1-8: Typ zařízení PDA, zdroj: Microsoft webcast

Pocket PC 2003

Tuto platformu není třeba příliš představovat. Je to přístroj s dotykovým displejem a několika dalšími ovládacími prvky. Prosadil se jako osobní digitální asistent. Rozšíření komunikačních možností, například o Wireless LAN, Bluetooth, GSM/GPRS docílíme pomocí externí karty (PCMCIA, CF nebo dokonce i SD). Novější přístroje mají dokonce některé z komunikačních modulů interní. Jestliže odbočíme od platformy Microsoft, nalezneme velmi podobná a úspěšná mobilní zařízení, která ovšem mají zcela jiný operační systém a jsou dodávány jinými firmami (Palm, Handspring). Nezávisle na faktu, že operační systémy těchto zmíněných zařízení jsou nekompatibilní, můžeme vyvíjet mobilní aplikace i pro tyto přístroje.

Zařízení jsou osazována procesory pracujícími na frekvencích řádově ve stovkách MHz. Interní paměť se pohybuje od 64 megabajtů. Displeje jsou barevné s rozlišením až 320x240 bodů. I tato třída zařízení většinou podporuje přídatné paměťové karty a další hardwarová rozšíření, stejně tak nabízí možnost připojení a synchronizace s osobním počítačem.

Pocket PC Phone Edition

Jednoduše konstatováno, přístroj Pocket PC 2003 Phone Edition je klasické PDA s integrovaným GPRS mobilním telefonem, jehož přítomnost je u starších přístrojů rozpoznatelná na první pohled podle anténky – novější přístroje již mají anténku integrovanou do těla přístroje, takže existenci mobilního telefonu v přístroji lépe poznáme dle slotu na SIM kartu nebo pochopitelně z technických údajů přístroje.

Smartphone 2003

Jedná se o mobilní telefon, přesněji o přístroj, jenž svojí konstrukcí spíše připomíná mobilní telefon, než-li PDA. Platforma Smartphone 2003 vychází v podstatě z Windows CE (Pocket PC), přičemž by měla kombinovat to nejlepší z mobilních telefonů a PDA. Tato platforma by měla poskytnout inteligentní spojení: pomocí hlasu, e-mailů nebo jiných prostředků, a co je nejdůležitější – aby spojení mohlo být navázáno kdekoliv a kdykoliv. Smartphone kromě elektronické pošty nabízí i možnost prohledávání webového obsahu. Využívá při tom takového uživatelského rozhraní, které je očekáváno od mobilního telefonu. Platforma Smartphone, jelikož je implementována v mobilních telefonech, má určité zvláštnosti – hlavním rozdílem od Pocket PC 2003 je absence dotykového displeje. Mobilní telefony se totiž ovládají hlavně pomocí klávesnice a ta je během přenosu zablokovaná, takže dotykový displej bez odklopného krytu by byl nesmyslem. Rozlišení displeje je zpravidla 176x220 pixelů. Displej může být barevný nebo černobílý.

Windows Mobile

Nedávno společnost Microsoft oznámila uvedení nového operačního systému Microsoft Windows Mobile 2003 pro Pocket PC. Nová obchodní značka Windows Mobile zastřešuje mobilní zařízení Smartphone, Pocket PC a Pocket PC Phone Edition. Windows Mobile umožňuje lepší orientaci v řadě softwarových produktů pro mobilní zařízení.

Produkty s označením Microsoft Windows Mobile přinesly zákazníkům i výrobcům mobilních zařízení následující výhody:

- Důvěrně známé prostředí.
- Vše v jednom. Microsoft vyvinul Windows Mobile tak, aby uživatelům pomohl maximálně zjednodušit život a práci s jednotlivými zařízeními, navrženými přesně podle přání a individuálních potřeb zákazníků.
- Být stále v kontaktu. Díky Microsoft Windows Mobile mohou zákazníci stále komunikovat s výkonnými a osvědčenými aplikacemi.
- Osobní produktivita. Využitím Microsoft Outlook, Internet Explorer, Windows Media Player a dalších aplikací uzpůsobených pro kapesní počítače uživatelé mohou zvýšit svoji produktivitu.

- Synchronizace na cestách. Microsoft Windows Mobile umožní aktualizaci důležitých dat jak při práci v kanceláři, tak na cestách.

Windows Mobile nabízí větší flexibilitu a podporuje další vývoj v oblasti hardware a software. Microsoft používá termíny Pocket PC, Pocket PC Phone Edition a Smartphone pro označení jednotlivých typů zařízení a značku Microsoft Windows Mobile pro označení operačního systému, který tato zařízení používají. Například z "Pocket PC software" se stává "Microsoft Windows Mobile software pro Pocket PC" a ze "Smartphone software" je "Microsoft Windows Mobile software pro Smartphone". Software pro mobilní zařízení se tak jednoduše stává operačním systémem Microsoft Windows Mobile.

1.2 Komunikace mezi vzdálenými zařízeními

V této kapitole si ukážeme možnosti komunikace mezi mobilními přístroji. Rozebereme jednotlivé architektury bezdrátových sítí, způsoby sdílení dat na dálku, současná řešení návrhu mobilních aplikací a vysvětlíme si pojem synchronizace dat.

1.2.1 Architektura bezdrátových sítí

Existují 2 druhy architektury bezdrátových sítí

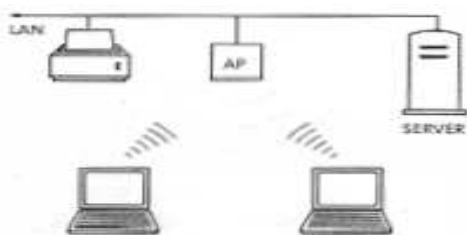
- Ad-hoc Networking
- Infrastructure Networking

Ad-hoc Networking

Ad-hoc Networking neboli Peer-to-Peer architektura je určena pro lokální síť s menším počtem počítačů. Umožňuje sdílení disků, souborů a přístup k Internetu přes sdílený modem. Výhodou je jednoduchá instalace a nízké náklady.

Infrastructure Networking

Při této architektuře, narozdíl od sítě nastavené jako Ad-hoc Networking, kde je tok dat ke každému počítači přímý, údaje přehazuje také takzvaný Access Point. Access Point může být hardwarový, nebo softwarový, vytvořený pomocí PC s wireless kartou a příslušným softwarem.



Obr. 1-9: Architektura Infrastructure Networking, zdroj Microsoft

Access Point obvykle umožňuje i přístup ke klasickým kabelovým sítím. Výhodou je zvýšení dosahu, protože každý připojený k rádiové síti má spojení v dosahu Access Pointu. Nevýhodou jsou vyšší náklady, zahrnující také hardwarový nebo softwarový Access Point. Základním problémem všech rádiových sítí je bezpečnost dat. Aby se zabránilo úniku informací, například odposlechům mimo budovu, je možné data šifrovat pomocí systému Wired Equivalent Privacy – WEP. WEP používá k šifrování dat 40 bitové klíče, které jsou identické u všech klientů v síti.

1.2.2 Možnosti sdílení, techniky přenosu dat

S příchodem mobilních technologií mohou uživatelé odkudkoli přistupovat k firemním datům – jako jsou například skladové zásoby, obsah emailové schránky, kalendář nebo jiné sdílené informace. Bez mobilního přístupu k databázím by většina z těchto vlastností byla nemyslitelná.

Jak to běžně celé probíhá? Pracovníci v terénu zadávají pomocí mobilních zařízení, ať už to je notebook nebo PDA nebo dokonce mobilní telefon, data týkající se právě prováděných operací, což dalším pracovníkům přináší kvalitnější přehled o aktuálním stavu procesů a dochází tak k optimalizaci průběhu firemních procesů.

U mobilních řešení rozeznáváme dva základní přístupy, jak zajistit sdílení dat na dálku, tzv. online a offline řešení.

Online řešení spočívá v navázání spojení mobilního zařízení s centrálním systémem. Při tomto typu řešení se předpokládá trvalé připojení mobilního přístroje k síti. Na mobilní zařízení je nainstalována jednoduchá aplikace, jenž umí prostřednictvím .NET remotingu (viz. 3.1.1) nebo webových služeb (viz. 3.1.2) komunikovat s databázovým serverem. Při provedení každé operace se mobilní zařízení spojí (přes internet nebo pomocí jiného bezdrátového připojení) s databázovým serverem, jenž je umístěn někde v kanceláři firmy. Databázový server provede operaci, o níž ho klient žádá. Výhodou tohoto řešení je okamžitý aktuální stav dat na serveru a na klientovi. Nevýhodou je relativně pomalá komunikace mezi klientem a serverem a podmínka trvalého připojení k síti.

Typická architektura tohoto řešení se skládá ze 2 vrstev:

- vrstva Front-end řeší interakci s uživatelem aplikace běžící na mobilním zařízení
- vrstva Back-end řeší komunikaci a datové toky uvnitř firmy (na straně serveru).

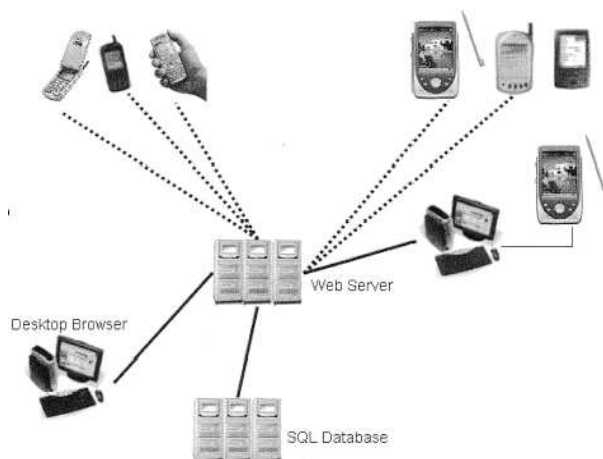
Rádiová síť podstatně zvýší efektivitu vrstvy Back-end. Tato vrstva se skládá ze serverových, transakčních a databázových komponent.

Oproti tomu offline řešení nevyžaduje pro práci s aplikacemi trvalé připojení k serverové části systému. Je tedy nutné, aby na mobilní zařízení byla nainstalována mobilní databáze, která představuje "zmenšený obraz" centrální databáze. Jinou možností je využití offline aplikačních bloků, které umožňují práci s cache. Mobilní pracovníci pak mohou kdykoli a kdekoli pracovat s mobilním systémem bez spojení s centrální aplikací a čas od času se "připojit" k centrálnímu systému a provést tzv. synchronizaci, která obnáší především aktualizaci dat na obou stranách, tj. v centrální databázi i v mobilní databázi. Aplikační blok synchronizaci řeší sám, sám identifikuje stav připojení a podle toho pracuje s daty buď offline (cache) nebo online (pomocí webových služeb).

Všechny mobilní zařízení mají některé z uvedených charakteristik :

- Využití lokálních zdrojů
- Využití síťových zdrojů
- Podpora občasného přístupu uživatelů
- Záruka flexibility klientských zařízení

Většina aplikací nepotřebuje všechny tyto charakteristiky. Při návrhu aplikace pro mobilní zařízení se musíme dopředu rozhodnout, na jakém principu bude naše aplikace pracovat, které z výše uvedených řešení online nebo offline je pro tuto aplikaci vhodné.



Obr. 1.10: Princip komunikace pomocí „tenkého“ klienta (on-line řešení), zdroj: Microsoft



Obr. 1.11: Princip off-line komunikace, zdroj: Microsoft

Kritéria výběru správného řešení

Při zavedení databázového serveru na určitou hardwarovou platformu nás především zajímají možnosti této platformy, hlavně paměťová kapacita a výpočtový výkon. U platformy Pocket PC lze výkon procesoru hodnotit celkem dostatečně, horší je to s kapacitou paměti pro ukládání dat. Proto je velmi důležitý správný návrh vytvářené mobilní aplikace.

Především nás musí zajímat nároky zákazníka. Pokud zákazník vyžaduje hlavně rychlý přenos dat, nechce investovat do připojení mobilních zařízení k internetu a postačuje mu, když se data budou aktualizovat jednou za čas, pak musíme volit offline řešení s možností synchronizace. Tím se ovšem zvyšují nároky na samotné mobilní zařízení. Musí být dostatečně výkonné, jelikož všechny operace se budou vykonávat lokálně. Musí mít dostatečnou paměťovou kapacitu, jelikož zařízení bude spolupracovat s lokální mobilní databází. Nesmí být také opomenuta dobrá bezpečnost dat na přístroji.

Jiným řešením je použití aplikačních bloků, které sice pracují v režimu offline, ale celá logika aplikace (business a datová vrstva) je uložena na serveru. Není potřeba mít lokální mobilní databázi na serveru a řešit stav připojení k síti. Aplikace vše vyřeší sama a podle stavu připojení pracuje buď s „živými“ daty ze serveru nebo s daty z cache.

Pokud ovšem zákazník netrvá na rychlosti a zaručí trvalé připojení všech klientských zařízení k síti, k tomu si zakládá na trvale aktuálních datech na obou stranách (klient i server), pak je řešením online struktura aplikace. Výhodou tohoto řešení je fakt, že se snižují nároky na mobilní zařízení, kde se nainstaluje jednoduchá aplikace. Nemusí se psát synchronizace dat a bezpečnost zajišťujeme pouze při přenosu dat mezi serverem a klientem (předpokladem je, že databázový server je dobře zabezpečený).

1.2.3 Synchronizace dat

Při návrhu klientské aplikace pro mobilní zařízení postavené na offline řešení (viz kapitola 1.2.2) se data ukládají do lokální mobilní databáze umístěné přímo na klientském přístroji. Nesmíme ale zapomenout na aktualizaci dat. Udržování aktuálních dat jak na serveru (databázový server), tak i na klientovi (například PDA) zajišťuje synchronizace.

Synchronizace slouží k přenášení dat mezi serverem a externími zařízeními, jako jsou kapesní počítače nebo mobilní telefony. Výsledkem přenosu je, že na obou místech jsou stejná a úplná data.

Pokud jde o způsoby synchronizace, existují v zásadě dva. Úplná (někdy též označovaná jako pomalá – Slow Sync) synchronizace a obousměrná (Two-Way Sync) synchronizace.

Úplná synchronizace – řádek po řádku porovná všechny záznamy na straně serveru i klienta a sjednotí data na obou stranách. Při tom nedochází k mazání žádných záznamů. Tato synchronizace proběhne vždy při prvním spuštění, anebo pokud je proces přerušen nestandardním způsobem. Pokud se synchronizuje větší množství objektů, může proces trvat několik minut. Pomalou synchronizaci lze také vynutit nastavením na serveru a někdy i volbou v mobilním zařízení.

Obousměrná synchronizace – běžný typ synchronizace, kdy si klient a server vymění informace o změnách ve svých datech. Spouští se z externího zařízení. Server porovná svoje data s daty klienta, sjednotí je u sebe a pošle je zpět klientovi. Tato synchronizace vychází z toho, že server má uložen stav obou databází, proto se při porovnávání vymazávají záznamy, které byly od poslední úspěšné synchronizace na jedné nebo druhé straně odstraněny.

Více informací o možnostech synchronizace dat mezi klientskou a serverovou databází najdete v kapitole 3.2

1.2.4 Bezdrátové připojení PDA k síti

Abychom mohli použít online řešení návrhu aplikace pro mobilní zařízení, kdy klientská aplikace je trvale připojená k síti a má tak možnost v reálném čase komunikovat se serverem (databází na serveru), musíme vymyslet způsob, jakým zařízení, na němž poběží klientská aplikace, připojit k síti.

Existuje několik možností připojení mobilního zařízení k síti.

Bluetooth

Bluetooth je název komunikační technologie definované konsorciem firem, které se zabývají bezdrátovou komunikací mezi zařízeními na krátkou vzdálenost. Bluetooth se využívá pro stolní počítače, přenosné počítače, personální digitální asistenty (PDA), počítačovou

periférii, mobilní telefony a jiná domácí a technologická zařízení. Komunikace probíhá duplexně ve volném kmitočtovém pásmu 2.45 GHz s dosahem 10 až 15 metrů při přenosové rychlosti 712 Kbps až 2 Mbps.

Nevýhodou tohoto způsobu připojení k síti je dost malá vzdálenost, na kterou mohou zařízení pomocí Bluetooth komunikovat (max. 20m). Proto je tento způsob vhodný například pro aplikaci typu restaurační zařízení (podrobný popis a ukázkou naleznete v této diplomové práci níže)

Wireless LAN

Lokální počítačová síť LAN spojuje dva nebo více počítačů a umožňuje sdílet disky, soubory, periférie anebo síťové služby. Obecně počítače propojuje strukturovaná kabeláž. Mobilní zařízení jsou bytostně spojena s bezdrátovou sítí. Jen tak může být zajištěna skutečná mobilita. Při bezdrátovém přenosu jsou informace přenášeny éterem, v tomto případě prostřednictvím rádiového přenosu v pásmu 2,4 GHz rychlostí 11 Mbps.

Tento způsob je vhodný pro návrh mobilních aplikací. Pokud jednotlivá zařízení „vidí“ na Access Point (přístupový bod) sítě, nemusíme mít trvalé připojení mobilních zařízení k internetu, komunikace bude rychlejší než přes http a práce efektivnější.

Mobilní telefon podporující protokol WAP

Definice protokolu popisuje datovou komunikaci mezi mobilním zařízením a sítí mobilního operátora. Jako ekvivalent internetových prohlížečů pro PC (Internet Explorer, Netscape Navigátor, Opera...) byl navržen prohlížeč pro bezdrátové aplikační prostředí WAE (Wireless Application Environment). Prohlížeč je založen na principu klient/server architektury a „tenkého“ klienta. Mobilní telefon musí mít poměrně velký display a vhodně navržené ovládací prvky pro surfování. Typické rozlišení displejů mobilních telefonů je asi 90x60 pixelů. Takové rozlišení displeje samozřejmě na zobrazení klasické HTML stránky nestačí. Praktické minimum pro zobrazení HTML stránky, včetně grafiky, zřejmě představuje displej handheldu, který má rozlišení 240x320 pixelů. Proto musí být webové stránky přenášeny pomocí protokolu WAP na mobilní telefony navrženy dle rozlišovací schopnosti zobrazovacího zařízení.

Klasické připojení zařízení k internetu

Mobilní zařízení můžeme připojit k internetu. Způsobů je několik. Níže je stručný přehled technologií.

GPRS je zkratka z anglického termínu "General Packet Radio Service" (v češtině „Obecná paketová rádiová služba“). Rychlost spojení závisí na schopnostech mobilního přístroje (maximální reálné hodnoty jsou pro downlink 14,4 kbit/s a pro uplink 43,2 kbit/s), operátorovi je placeno pouze za přenesená data, nikoliv za dobu spojení.

EDGE je zkratka z anglického termínu "Enhanced Data for GSM/GPRS Evolution (v češtině „Rozšířená data pro rozvoj GSM/GPRS“). Princip fungování je stejný jako u GPRS, ze kterého také EDGE vychází. Proti němu EDGE nabízí několikrát vyšší rychlosti. Velkou výhodou EDGE je, že pokud je telefon podporující EDGE, je přecházení mezi EDGE a GPRS naprosto plynulý a transparentní. Uživatel nic nepozná, tedy kromě někdy i citelného zvýšení/snížení přenosové rychlosti.

UMTS je zkratka z anglického termínu "Universal Mobile Telecommunication System", (v češtině „Univerzální mobilní telekomunikační systém“). V principu jde o v tuto dobu nejrychlejší mobilní datové připojení (WiFi nepočítaje). Pro komunikaci potřebujeme zařízení s UMTS modulem, se systémem Windows Mobile je na trhu zatím jen jediné - HTC Universal. Nabízená rychlost je 50KB/s. Na druhou stranu může být pro většinu uživatelů problémem pokrytí, které rozhodně není úplné.

HSCSD je zkratka z anglického termínu "High Speed Circuit Switched Data", což je systém GSM založený na technologii přepojování okruhů. Důležité jsou dvě skutečnosti – maximální rychlost je 57,6 kbit/s a operátorovi je placeno za čas spojení.

2 Technologické možnosti

V této kapitole probereme technologické možnosti pro vývoj aplikací pro mobilní zařízení. Popíšeme platformu, na které budeme psát zdrojový kód a navrhovat strukturu aplikace. Také se zaměříme na vývoj mobilních aplikací z historického hlediska.

2.1 Terminologie

Pro získání orientace v pojmech, jako je operační systém Windows CE, WindowsCE.NET, platforma Pocket PC, Smartphone a podobné, zavedeme určité terminologické konvence. V první řadě je potřeba rozlišovat mezi operačním systémem a platformou.

Operační systém je hlavní řídicí program jakéhokoliv počítače. Jeho úlohou je zpřístupnit hardwarové porty a zařízení (displej, klávesnice...) jednotlivým aplikacím a jejich prostřednictvím i uživateli. Je to první program, který byl do tohoto počítače zaveden z určeného paměťového média při jeho spuštění. Obvykle se operační systém skládá z jádra a obsluhy zařízení. Typickým příkladem operačního systému je například Windows CE, Windows CE.NET, Windows XP, Windows XP Embedded, MacOS, Linux, OS/2 a další.

Platforma je všeobecně definována jako určitý hardware podporovaný množinou programů, modulů, komponent pro styk uživatele a operačního systému. Například PocketPC, PocketPC 2003, Palm, Smartphone a TabletPC, desktopy a notebooky typu PC.



Obr. 2.1: Typy mobilních přístrojů a jejich platformy, zdroj: Microsoft

Je potřeba zvolit typ a způsob mobilní komunikace a také typ prostředků pro vykonání komunikace. Na typu vybraných prostředků komunikace pak závisí i způsob komunikace.

2.2 Platforma pro vývoj

.NET

Platforma Microsoft .NET (čteno dot-net) je postavena na mohutném rozvoji využívání počítačů prostřednictvím internetu a pro komunikaci.

Platforma .NET usnadňuje průběžné dodávání softwaru uživatelům prostřednictvím distribuovaného počítačového modelu pro internet, který unikátním způsobem využívá v maximální míře komunikace i počítačové technologie.

Platforma .NET je postavena na internetových protokolech a standardech pro interakce mezi zařízeními a službami a zejména na Extensible Markup Language (XML).

Oblíbené marketingové tvrzení o tom, co je to Microsoft .NET, říká, že se jedná o „platformu webových XML služeb obsahujících vše, co je nutné pro tvorbu a používání aplikací založených na XML, společném jazyku pro výměnu dat“. Co si ale pod touto skoro nic neříkající větou má představit počítačový laik? Je to vývojový nástroj? Je to operační systém? Nebo metodologie? Služby? Odpověď zní: Od každého trochu, .NET je všechno z toho dohromady.

.NET je platforma pro běh nových distribuovaných aplikací. Jelikož jsou jakákoli řešení postavená na .NET vlastním programováním (tedy ne skriptováním, které se používá například v technologiích typu ASP/PHP), říká se jim aplikace.

Aplikace postavené na .NET platformě budou mít stejné rysy a stejný model návrhu, ať už jsou to aplikace pro přenosná zařízení typu Pocket PC, aplikace pro desktopové použití, pro velmi výkonné serverové produkty či – což je pro nás nejdůležitější – tvorbu webových stránek.

Srdcem platformy .NET je jádro nazvané .NET framework. Toto jádro se stará o tři základní úkoly:

- Je run-time prostředím pro běh aplikací, tzn. aplikace pro něj jsou s ním pevně spjaty a bez něho ani nejdou spustit
- Poskytuje možnosti pro tvorbu ASP.NET stránek i webových služeb
- Obsahuje velmi širokou řadu knihoven, objektově orientovaných tříd usnadňujících řadu úkolů (bezpečnost, komunikaci, práce s databázemi a datovými zdroji)

Výhody platformy .NET:

- Verzování knihoven a digitální podpis knihoven
- Stejný programový model pro programování ve Visual Basic.NET, C++.NET nebo C#.NET
- Možnost využití nových programovacích jazyků na platformě .NET. Například použití kompilátorů pro Perl, Python či Fortran.
- Vylepšené mechanismy správy paměti – Garbage Collector monitorující paměť zdroje použitého v aplikaci a vracející tu paměť dle potřeb zpět systému
- Ošetřování chyb. Chyby se zpracovávají pomocí výjimek a jejich následného zpracování
- Jazyky postavené na .NET jsou typově přísné.

Architektura .NET rozdělila mechaniku aplikací do dvou fází. Nejdříve se napsaný program zkompile pomocí kompilátoru do tzv. Intermediate Language (dále IL). Jako programovací jazyk je možné použít Visual Basic .NET, C#, C++, JScript od Microsoftu nebo některý z plánovaných jazyků třetích stran, připravuje se např. FORTRAN, Python a Perl a mnoho dalších. Přeložené IL instrukce v binární podobě jsou uloženy přímo v .exe nebo .dll souboru.

Při spuštění aplikace se IL nejdříve přeloží Just In Time (JIT) pomocí kompilátoru do strojového kódu a ten je následně proveden.

.NET Compact Framework

Jak napovídá samotné jméno, je Compact Framework ořezanou verzí .NET Framework firmy Microsoft. Tento produkt je určen pro mobilní a vestavná zařízení jako je Pocket PC nebo Windows Powered Smartphone. Vyvíjíme-li aplikaci pro tato mobilní zařízení, poskytuje nám .NET v podstatě dvě možnosti. Vytvořit aplikaci v prostředí Compact Framework a nebo ASP.NET Mobile Control (bývá také označován jako Microsoft Mobile Internet Control). ASP.NET Mobile Control je řešení postavené na architektuře klient-server. Nevýhodou tohoto řešení je nutnost neustálého připojení k síti a také nemožnost využití dalších periferních zařízení. Naproti tomu v prostředí Compact Framework lze tvořit plnohodnotné aplikace. Jak již bylo zmíněno Compact Framework je podmnožinou .NETu. Neobsahuje tedy například žádné specifické jmenné prostory. Compact Framework se skládá ze dvou hlavních částí: vývojového prostředí a prostředí pro běh aplikace.

Vývojové prostředí pro vytváření aplikací pro mobilní zařízení je přímo integrováno do Visual Studia 2003 a nově do Visual Studia 2005. Vývojové prostředí, označované jako Smart Device Extension, je jeden z typů projektů, které lze v prostředí VS vytvořit, vyvíjet a ladit. Vytvořená aplikace je pak spouštěna na Compact Framework Common Language Runtime (CF CLR). Mobilní zařízení proto musí implementovat toto běhové prostředí. Novější zařízení jej podporují přímo. U ostatních je nutné toto prostředí doinstalovat. Hlavní vlastnosti Compact Runtime CLR jsou stejné jako u běhového prostředí plnohodnotného .NETu. Mezi tyto vlastnosti patří typově bezpečné běhové prostředí, garbage collection, just-in-time kompilace, ošetření chyb pomocí výjimek, společný typový systém.

V několika bodech se ale liší. Podstatným rozdílem je vypuštění některých knihoven (CF .NET jich obsahuje asi čtvrtinu). Z dalších rozdílů uvedme například absenci binární serializace, ASP.NET, podpora kompilování aplikace v době instalace (install-time JIT), podpora pro psaní COM/ActiveX objektů.

Výsledné okleštěné běhové prostředí má okolo 2MB.

2.3 Stručná historie vývoje aplikací pro mobilní zařízení

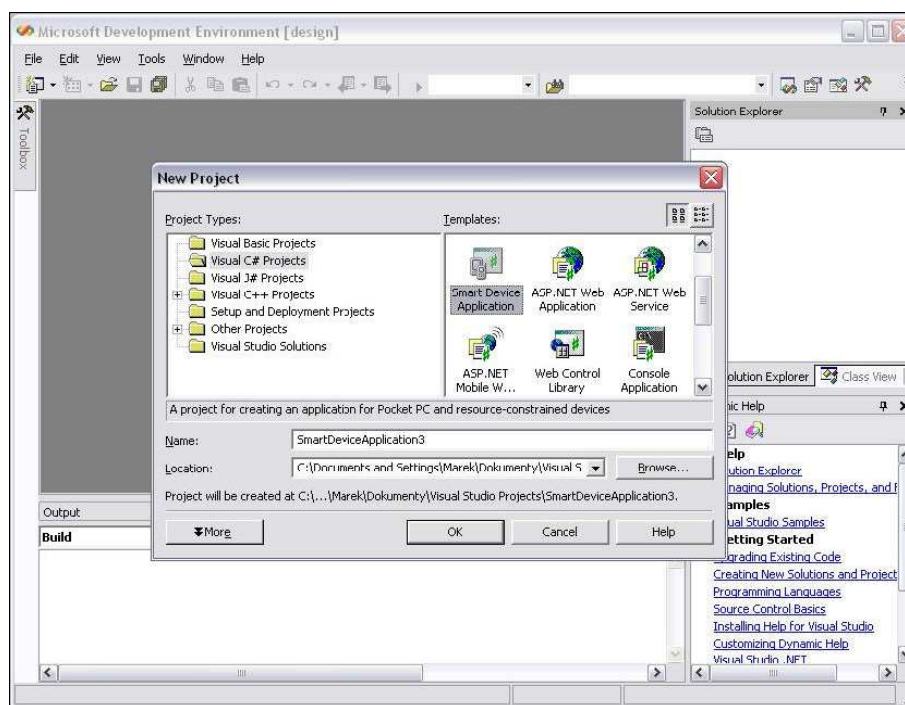
První verze .NET Frameworku s první verzí vývojového prostředí Visual Studio .NET vývoj mobilních ASP.NET aplikací nepodporovala. Brzy se však objevila beta verze nástroje Microsoft Mobile Internet Toolkit (MMIT). Tato verze nikdy nebyla komerční, jelikož pod názvem ASP.NET Mobile Controls byla přímo implementována do .NET Frameworku 1.1 a následující verze vývojového prostředí Visual Studio .NET 2003. Pro vývoj a testování aplikací pro platformu Pocket PC je také zapotřebí nainstalovat emulátor Pocket PC. Můžeme využít emulátor z volně šiřitelného vývojového prostředí eMbedded Visual Tools. Pro emulování WAP telefonů můžeme použít například OpenWave emulátor. V novější verzi ASP.NET stránek byla přijata nová koncepce. Místo speciálních ovládacích prvků pro vytváření formulářů určených pro mobilní zařízení je možné použít též některé stejné prvky jako pro klasickou aplikaci.

Na základě rozpoznání typu klientského prohlížeče webových stránek a požadavku na vygenerování stránky se zapojí do hry některý z adaptérů, nejčastěji HTML adaptér nebo WML adaptér pro WAP stránky a vyrenderuje příslušný typ stránky. Tuto technologii implementovanou v ASP.NET 2.0 nazýváme adaptivním rendrováním.

2.4 Vývoj aplikací pro mobilní zařízení na platformě .NET

PDA má mnohem větší možnosti než většina ostatních mobilních zařízení, jako jsou např. mobilní telefony. Jde zejména o velikost paměti, rychlost zpracování instrukcí a v neposlední řadě také o velikost zobrazovací plochy. První, co je třeba při tvorbě aplikace provést, je zvolit prostředí (platformu), na níž bude celá aplikace založena. Volba prostředí bude záviset na operačním systému PDA.

Při návrhu a vývoji aplikace v prostředí .NET je k dispozici nástroj MS Visual Studio. Zde lze celkem rychle a přehledně tvořit aplikace všeho druhu. Jedná se o kompletní prostředí s editorem zdrojových textů, form editorem, debuggerem a kompilátorem. Při tvorbě aplikace je třeba vybrat nejprve typ aplikace (resp. typ celého projektu). Pokud to má být aplikace pro PDA, stačí vybrat typ projektu Smart DeviceApplication.



Obr. 2.2: MS Visual Studio

Je zapotřebí zvolit typ aplikace. Zda má být pro Pocket PC nebo Windows CE, což jsou zřejmě dva nejdopodporovanější operační systémy a o jaký druh aplikace se jedná (Windows application, class library, non-grafical application). Pak už nezbyvá než aplikaci naimplementovat. K ulehčení práce s formuláři je k dispozici také form editor, kde lze „naklikat“ kompletní grafické rozhraní aplikace. Poklepáním na některý z prvků formuláře se zobrazí textový editor, kde je možné dopsat kód k danému prvku, který se provede, když nastane příslušná událost, jako kliknutí na tlačítko a podobně.

Výslednou aplikaci je možné spustit v emulátoru nebo přímo nainstalovat a spustit v PDA. Emulátor kompletně napodobuje operační systém Pocket PC (resp. Windows CE) a je v něm možné provádět stejné operace jako na PDA.

Každý informační systém poskytuje informace. Téměř vždy jde o architekturu klient-server, kdy uživatel (resp. aplikace) v roli klienta přistupuje do informačního systému. Server převezme požadavek, zpracuje jej a předá zpět požadované informace. Klientská aplikace zde může figurovat buď jen jako zobrazovač přijatých dat (s minimálními funkcemi) nebo jako aplikace zapouzdřující nějakou logiku či nějaký systém pro uchovávání dat. Komunikace se serverem spočívá jen ve výměně dat nebo aktualizaci svých dat. V prvním případě hovoříme o tenkém klientovi (thin-client) a v tom druhém o tlustém klientovi (thick-client).

3 Definování toku dat v obou směrech komunikace a datového formátu

V této kapitole si vysvětlíme způsoby přenosu dat mezi klientem a serverem. Popíšeme, jak takový přenos funguje a v jakém formátu se data přenášejí. Nakonec uvedeme ukázkovou aplikaci, která slouží pro restaurační zařízení a umožňuje online práci s daty na serveru.

3.1 Možnosti komunikace mezi mobilním zařízením a databázovým serverem

Jak již bylo popsáno v předchozích kapitolách, existují dva způsoby návrhu aplikací pro mobilní zařízení, schopných komunikovat na dálku: online a offline. Pro návrh aplikace v režimu online musíme umožnit aplikaci „napřímo“ komunikovat s databází na serveru. V reálu to samozřejmě „napřímo“ nefunguje, jelikož databáze běží na vzdáleném serveru a pro přístup k ní se využívá různých technik. A právě techniky přenosu dat na dálku budou popsány v následujících podkapitolách. Pro práci v režimu online ukážeme dvě možnosti: .NET remoting a webové služby.

Jde o soubor technik a služeb, které umožňují komunikaci mezi dvěma a více vzdálenými objekty. Pod pojmem vzdálený objekt rozumíme především to, že objekt komunikuje s jinými objekty za hranicemi své aplikační domény, v níž je umístěn. Aplikační doména izoluje aplikaci v ní běžící od okolí (jiných aplikačních domén).

Komunikace mezi objekty z různých aplikačních domén je vzdálenou komunikací. Objekt, který v rámci této komunikace voláme, je vzdálený (serverový) objekt. Volajícímu objektu obecně říkáme klient (klientský objekt). Vzdálený objekt může být vůči klientovi v jiné aplikační doméně jiných procesů, v aplikační doméně jiného procesu na témže fyzickém stroji, nebo v jiné aplikační doméně jiného procesu na jiném fyzickém stroji (kdekoliv v síti LAN, WAN). V režimu práce offline se data lokálně ukládají na klientovi a jednou za čas jsou synchronizovány se serverem. Tento způsob přenosu dat popíšeme v kapitole 3.1.3.

3.1.1 .NET Remoting

Architektura systému

Distribuované systémy jsou vždy systémy typu klient/server. .NET remoting umožňuje navázat spojení mezi klientem a serverem.

.NET platforma umožňuje volat objekty pouze lokálně, tedy v rámci jedné aplikační domény. Vzdálený objekt musí splnit určité požadavky k tomu, aby byl vzdáleně volán. Procesu uschopnění vzdáleného objektu k volání klientem říkáme *marshaling*.

Marshaling může být dvojího typu: marshalig odkazem (by reference MBR) a hodnotou (by value MBV).

Marshaling hodnotou spočívá v serializaci vzdáleného objektu do proudu dat, jenž se následně deserealizuje (rekonstruuje) na klientovi. Podstata tohoto typu marshalingu je v tom, že se na klientovi vytvoří kopie serverového objektu. Pro více klientů se vytvoří více kopií serverových objektů (pro každého klienta svůj) se svými vnitřními stavy. Každý klient pak volá svojí kopii objektu lokálně. Vzdálený objekt musí být serealizovatelný, v kódu pak musí být buď označen atributem `Serializable` nebo implementovat rozhraní `ISerializable`. Výhoda marshalingu hodnotou je v rychlosti volání objektu, zvlášť pokud je objekt volán několikrát přes síť. Nevýhodou je však nemožnost serializovat velké objekty (zatížení sítě), nebo také použití objektů, jejichž správná funkce je závislá na jedinečnosti stroje, na kterém jsou spuštěny (např. adresářová struktura, databázové spojení). Nevhodné je také serializovat objekty, kompletní strukturu, které nebudeme potřebovat (např. u objektu zaměstnanec ne vždy potřebujeme vědět všechny informace, postačí nám např. pouze jméno a telefonní číslo).

Technika marshalingu odkazem (by reference) spočívá ve vytvoření proxy objektu v aplikační doméně klienta. Proxy objekt je speciální objekt, který je navenek pro volající objekt stejný jako serverový objekt. Proxy objekt ví o všech metodách, událostech, rozhraních a parametrech vzdáleného objektu, ale neumí volání klienta obsloužit sám. Jednotlivá volání klienta předává serverovému objektu a poté předává návratové hodnoty ze serveru zpátky klientovi. Klient neví o tom, že nevolá přímo vzdálený objekt. Vzdálený objekt musí dědit ze třídy `MarshalByRefObject` nebo z jejího potomka.

Proxy objekt

Proxy objekt se vytvoří vždy když klient aktivuje vzdálený objekt typu `MarshalByRef`. Tento objekt odkazuje na vzdálený objekt a zároveň je řízen správcem paměti (Garbage collector) stejně jako všechny objekty. Proxy objekt se skládá ze dvou objektů: `TransparentProxy` a `RealProxy`. `TransparentProxy` komunikuje přímo s klientem, respektive klient komunikuje s proxy objektem. Zvenku vypadá stejně jako vzdálený objekt. `TransparentProxy` vezme ze zásobníku volání vzdáleného objektu, převede je na tvar `IMessage` a přepoše objektu `RealProxy` tím, že zavolá jeho metodu `Invoke()`. `RealProxy` objekt je zodpovědný za přenos zprávy vzdálenému objektu. `TransparentProxy` je vytvořen z interní stejnojmenné třídy a je vlastněn objektem `RealProxy`. .NET platforma podporuje dva typy kanálů: tcp a http (třídy `TCPChanel` a `HTTPChanel`). Mohou být samozřejmě používány i kanály založené na jiných protokolech, např. SMTP a Jabber. Kanál http se většinou používá v prostředí internetu ve spojení s webovým serverem (IIS) a kanál TCP v prostředí intranetu. Přenášení zpráv kanálem TCP je výkonnější kvůli binárnímu formátovači. Jeho výhodou je to, že je spojově orientovaný.

Kanály a jejich funkce

Přenášení zpráv mezi klientem a serverem je realizováno přes kanály. Kanál je objekt, jenž se skládá s klientské a serverové části. Klientská část kanálu posílá požadavky na služby vzdálenému objektu, případně vrací návratové hodnoty. Serverová část naslouchá na příslušném portu na přicházející požadavky. Kanál musí být založen jak na serverové tak i na klientské straně ještě před zahájením volání vzdáleného objektu.

Kanál je složen z několika částí, na vstupu je formátovač, jenž serializuje zprávu na proud bajtů. Při formátování se provádí transformace zpráv. Druh transformace (formátovače) odpovídá většinou druhu kanálu. Pokud například používáme http kanál, tak defaultní formátovač bude typu SOAP (třída SOAPFormatter). Dále se zpráva předá dalším částem kanálu, pokud existují. Zde je možné zprávu zašifrovat. Poslední částí kanálu je TransportSink. TransportSink je část kanálu, která je zodpovědná za fyzický přenos mezi aplikačními doménami. Složení kanálu na straně serveru je stejné, akorát je v obráceném pořadí.

Jak na straně zákazníka, tak na straně klienta může být zaregistrováno více kanálů. Kanály je možné manuálně odregistrovat. Automaticky se odregistrování kanálu provede po ukončení nebo pádu aplikace. Při registraci kanálu se vytvoří takzvaný endpoint (jmenné mapování vzdáleného objektu pomocí URI) a port, na kterém kanál naslouchá a zasílá data. Port, na kterém kanál registrujeme, nesmí být obsazen (v rámci fyzického stroje, ne aplikační domény). Každý kanál musí implementovat rozhraní IChannelReceiver (serverová část) nebo IChannelServer (klientská část) nebo oboje dohromady. Záleží na tom, zda má být kanál obousměrný nebo jednosměrný.

Vzdálené objekty

Serverová aplikace je hostitelem vzdálených objektů. V .NET remotingu musí být každý vzdálený objekt vždy přisouzen nějakému hostiteli. Může to být konzolová aplikace, služba Windows, webová aplikace či klasická WinForm aplikace.

Vzdálené (serverové) objekty musí být buď server-activated (dále jen SA) nebo klient-activated (dále jen CA). SA objekty jsou známy pod pojmem well-known objekty. Mohou pracovat v jednom ze dvou módů: Singleton nebo SingleCall. Vzdálený SA objekt typu SingleCall je vytvořen při první žádosti (volání klienta) a existuje jenom jeden pro všechny klienty, tzn. umožňuje sdílení dat. Jelikož Singleton je připraven obsloužit jakéhokoliv klienta, musíme zabezpečit sdílení zdrojů pro vícevláknový přístup. Naproti tomu vzdálený SA objekt typu SingleCall se vytváří pro každé volání (požadavek) vzdáleného objektu. SA SingleCall vzdálené objekty neudržují žádné stavy mezi jednotlivými voláními a neumožňují sdílení dat mezi jednotlivými klienty. CA vzdálené objekty se vytváří na žádost klienta a pro každého klienta existuje pouze jedna instance. Stav mezi voláním u klienta je udržován, sdílení mezi klienty však nikoliv. Hlavní rozdíl mezi CA a SA objekty je v jejich aktivaci. Dřív než skutečn

volání vzdáleného objektu, musíme ho zaregistrovat ve službách .NET remoting. Registraci vzdálených objektů provádíme až po registraci kanálů. Pro registraci potřebujeme znát jméno assembly, ve které je definována třída objektu, dále endpoint, pomocí kterého je vzdálený objekt mapován a typ objektu. V případě, že aktivujeme SA objekt, potřebujeme vědět jeho aktivací mód (SingleCall nebo Singleton).

Aktivace vzdálených objektů

Volání vzdáleného objektu musí předcházet jeho aktivace. Pokud je vzdálený objekt marshalován hodnotou, musí se vytvořit a přenést jeho kopie ze strany serveru. Pokud objekt je marshalován odkazem, na klientské straně se musí vytvořit proxy objekt. Aktivaci objektu zastřešuje třída Activator. Ta nám poskytuje dvě důležité statické metody. Metoda CreateInstance() je určena pro aktivaci CA objektů. Při aktivaci se klient nejdříve spojí se serverem. Na straně serveru marshaluje vzdálený objekt a tento objekt se serializuje a pošle na stranu klienta. Tam se vytvoří buď kopie (pro vzdálené objekty zděděné z třídy MarshalByValue) nebo proxy objekt (pro vzdálené objekty zděděné z třídy MarshalByRef). Metoda GetObject() je určena pro aktivaci SA objektů. V tomto případě se u objektu marshalovaných odkazem nevytváří spojení mezi klientem a serverem, abychom mohli vytvořit objekt proxy. Aktivace SA objektu podporuje pouze bezparametrové konstruktory, protože se proxy objekt vytváří na straně klienta. Naopak u aktivace CA objektu, mohou mít tyto objekty konstruktory s parametry, neboť proxy objekt se vytváří na straně serveru. V obou případech potřebujeme vědět jméno stroje v síti, kde běží server, typ kanálu pro spojení, port, endpoint a samotný typ objektu. Všechny informace na straně klienta relativně známe, až na typ vzdáleného objektu. Máme několik možností zpřístupnění jména vzdáleného objektu na klientu.

- Implementace třídy vzdáleného objektu v .dll knihovně
- Definice rozhraní třídy vzdáleného objektu ve vlastní knihovně
- Definice a implementace třídy na straně hostitelské aplikace.

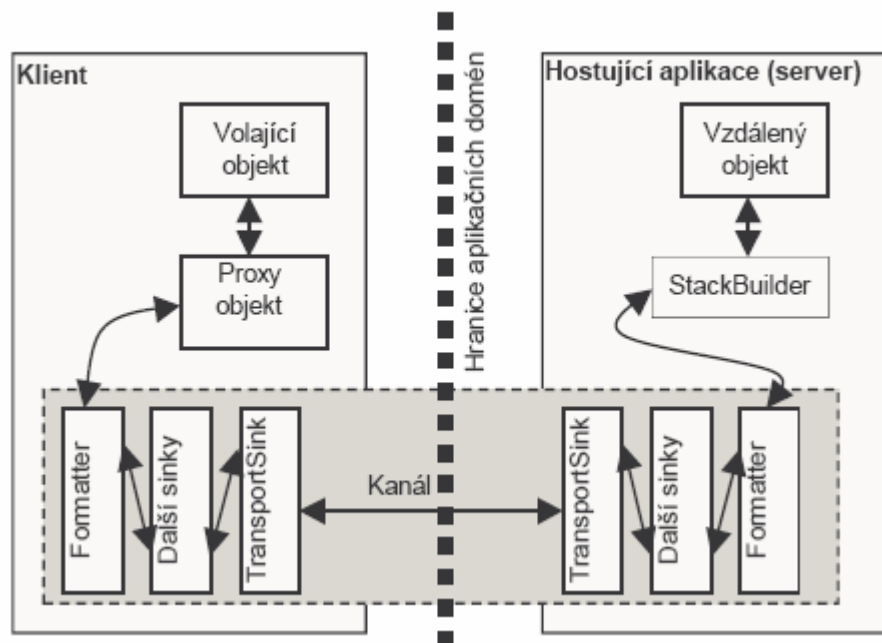
Proces tvorby remotingové aplikace

Na straně serveru

- zvolíme umístění vzdáleného objektu
- zvolíme hostitelskou aplikaci
- zvolíme typ kanálu a jeho port, zaregistrujeme kanál v .NET remotingu (kanálů může být více)
- zvolíme typ aktivace (případně mód vzdáleného objektu), endpoint a objekt zaregistrujeme ve službách remotingu

Na straně klienta

- zvolíme typ klienta
- zvolíme kanál, který následně zaregistrujeme
- aktivujeme vzdálený objekt
- voláme vzdálený objekt



Obr. 3.1: Volání vzdáleného objektu (MarshalByRef), zdroj: „NET remoting pod lupou”

3.1.2 Webové služby

Co jsou webové služby?

Webové služby (web services) jsou vlastně systémem pro podporu součinné spolupráce počítačů v síti. Pod tím si asi málokdo představí něco konkrétního. V podstatě jde o komunikaci mezi dvěma počítači, při níž má jeden funkci poskytovatele webové služby a druhý je klientem. Poskytovatel (provider) služby poskytuje data specifikovaným způsobem na síti. Na druhé straně si klient (requestor) zjistí adresu služby (vyhledá v registru nebo má adresu přímo od poskytovatele), stáhne si popis služby a je ji následně schopen využívat. Data se přenášejí v XML formátu a díky němu jsou snadno rozšiřitelná.

Praktické využití webových služeb

1. Propojení firemního intranetu se specializovaným produktem, který s daty pracuje. Intranetový informační systém funguje jako poskytovatel služeb. Specializovaný produkt z něj pomocí webové služby může čerpat data.

2. Nabízení ve fungujícím e-shopu produktů jiného výrobce. Partner (jiný výrobce) zpřístupní data o svých produktech pomocí webové služby a e-shop z té webové služby bude data číst, případně může provádět i jiné operace.
3. Stahování aktualizací do desktopové aplikace a rozšíření.
4. Využití datové a aplikační vrstvy běžící na straně serveru mobilní aplikace umístěné na mobilním přístroji nebo notebooku přes bluetooth, wi-fi nebo internet.

Použití webových služeb je velmi jednoduché. Veškerá data publikovaná dobře navrženou webovou službou je možné okamžitě dodat například obchodním partnerům. Není potřeba je přizpůsobovat pro každého zvlášť. Stačí poslat adresu služby. Webová služba je sama o sobě dost popisná a obecná. Také jde samozřejmě obecně použít i na procesy a metody.

Princip fungování webové služby

Proces publikace služby, její vyhledání a připojení musí být realizováno přesně definovanými protokoly. Ty lze rozdělit do 4 oblastí:

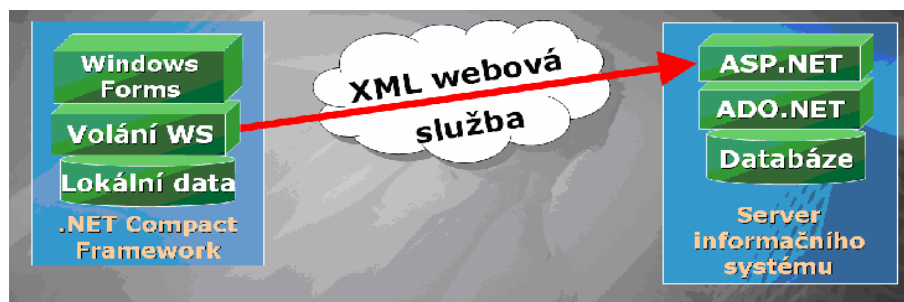
- přenos služby,
- XML komunikace (XML messaging),
- popis služby (Service description) a
- vyhledání služby (Service discovery).

Přenos služby

Přenos dat zajišťují standardní komunikační protokoly http, SMTP, FTP a další.

XML komunikace

Veškerá komunikace mezi poskytovatelem a klientem probíhá v XML zprávách. Komunikaci zajišťují protokoly, které je využívají. Nejčastěji jsou to XML-RPC, SOAP a REST. XML-RPC je starší protokol a SOAP je v podstatě jeho nástupce. SOAP pracuje na principu vzdáleného volání procedur (RPC). Poskytovatel tedy klientovi nabízí množinu funkcí nad daty. Klient pomocí XML tyto funkce vzdáleně volá.



Obr. 3.2: Schéma propojení klienta a serveru webovou XML službou, zdroj: Microsoft

Popis služby (service description)

Popis služby nám zajistí právě ony potřebné informace o rozhraní služby. Jedná se o popis komunikace, formát zpráv, názvy procedur, typy parametrů, vazby protokolu a další. Dnes se většinou používá jazyk WSDL (Web Services Description Language), který využívá SOAP pro komunikaci. Tento jazyk převzalo pod svá křídla W3C, v současnosti se dokončuje verze 2.0.

Vyhledání služby

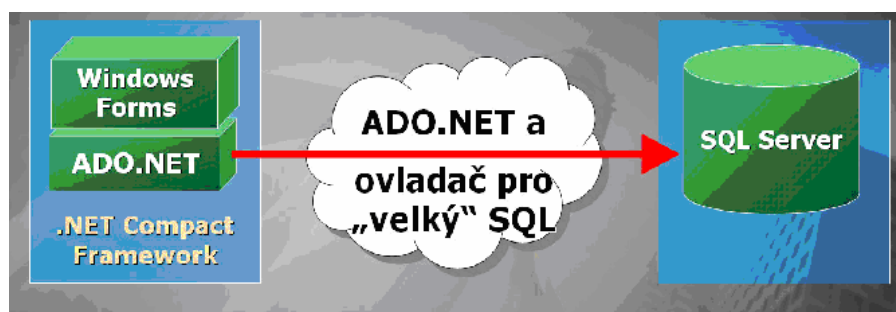
Pro vyhledání webové služby se běžně používá registr služeb, kam poskytovatel služby vloží její adresu a popis. Nejčastější implementací je UDDI (Universal Description, Discovery and Integration). UDDI je otevřený obchodní registr založený na XML. Nabízí seznam služeb a kontaktů, v kterých je možno vyhledávat.

3.1.3 Další způsoby připojení k serverové databázi

Uvedli jsme dva nejefektivnější způsoby sdílení dat na dálku: webové služby a remoting. Existují ale i další, nejsou tak efektivní a nebudeme je používat, nicméně přeci jen stojí za zmínku.

Přímý přístup k databázi

Pokud mobilní zařízení je připojené k síti, můžeme využít přímý přístup k „velké“ databázi na serveru. Stačí mít ovladač a znát příslušný connection string.



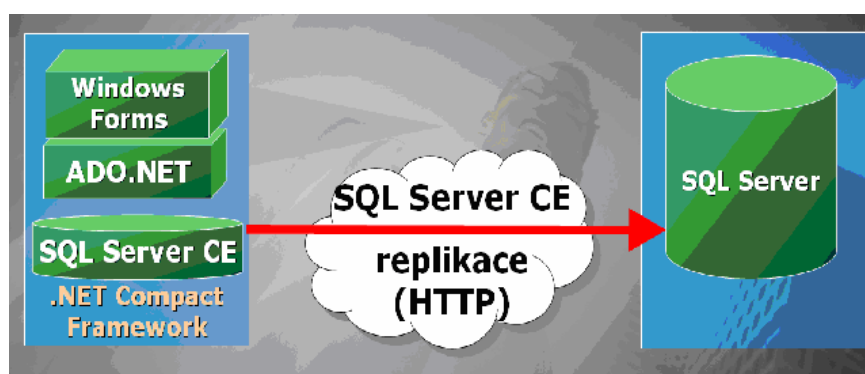
Obr. 3.3: Přímý přístup k databázi, zdroj:Microsoft

Mezi výhody takového přístupu patří jednoduchost řešení, malý objem dat na klientovi a nezávislost na mobilní databázi (můžeme využívat libovolně složité SQL příkazy a uložené procedury). Nevýhody jsou však podstatnější. Nejdůležitější je to, že v tomto případě neexistuje žádná střední vrstva, celá logika aplikace běží na klientovi. Serverová databáze musí být „přístupná zvenku“, což znamená velké bezpečnostní riziko v případě ukládání citlivých dat.

SQL Server Mobile a replikace

V tomto případě jde o synchronizaci dat mezi mobilní a serverovou databází. Klient ukládá data do lokální databáze, která je vždy přístupná. Pokud je připojen k síti nebo poté, až se k síti připojí, data se automaticky synchronizují. Tato technika je velmi propracovaná a nevyžaduje zvláštního programování. Komunikace pak probíhá přes protokol http. Více o synchronizaci dat bude uvedeno v kapitole 3.2.

Výhodou tohoto řešení je bezpečné úložiště dat jak na serveru tak na klientovi, možnost komplexního využití jazyka TSQL. Programátor může využít propracovaných technik synchronizace. Nevýhodou je nutnost konfigurace serveru a také ten fakt, že databázový engine zabírá nějaké místo v paměti mobilního zařízení.



Obr. 3.4: Synchronizace dat mezi „velkým“ SQL Serverem a mobilní databází pomocí replikací, zdroj Microsoft

Přístup k SQL Serveru přes HTTP

Příkaz v jazyku SQL může být i součástí URL adresy zadané z PC nebo mobilního zařízení prostřednictvím prohlížeče internetových stránek. SQL server potom vygeneruje údaje ve formě XML souboru.

Aby SQL server vygeneroval údaje ve formátu XML, používáme SQL příkaz doplněný o klauzuli *FOR XML AUTO*

Například *SELECT * FROM Employees FOR XML AUTO*

Před tím, než budeme moci přistupovat k databázi prostřednictvím SQL příkazu v URL adrese, musíme nakonfigurovat Internet Information Server pro podporu virtuálního adresáře. Pomocí nástroje IIS Virtual Directory Management for SQL server vytvoříme pro jakýkoliv fyzický adresář virtuální adresář. Povolíme služby, které budeme na virtuální adresář směřovat: URL queries, template queries, XPath, POST.

Samozřejmě, že musíme nastavit přístupová privilegia (pomocí nástroje SQL Server Enterprise Manager)

Ted' již můžeme prostřednictvím internetového prohlížeče zadáním URL adresy položit dotaz přímo na SQL Server, například takto:

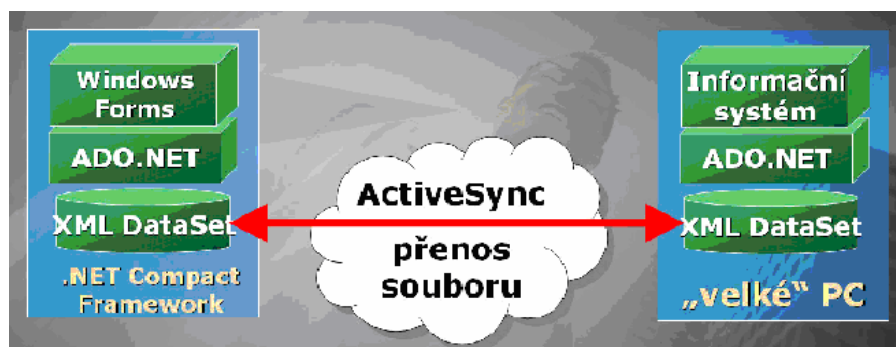
*http://llhome/vd_nwind?sql=SELECT*FROM Employees FOR XML AUTO & root=ROOT*

V okně prohlížeče uvidíme výpis údajů ve formátu XML.

Ne vždy se nám ale XML formát hodí. Proto je potřeba napsat XSL šablonu, ve které určíme formát výstupu údajů na HTML stránce klienta.

Replikace XML souborů pomocí ActiveSync

S touto koncepcí se setkávají uživatelé mobilních zařízení poměrně často. Jednak je to synchronizace souborů, na kterých se pracuje, například přenos elektronické knihy, dokumentů Word – Pocket Word, případně synchronizace dat mezi aplikacemi Outlook a Pocket Outlook. Výhodou je možnost práce offline, jen s občasným připojením se k desktopu nebo serveru. Určitou nevýhodou je současná možnost připojení jen jednoho zařízení k jednomu PC.



Obr. 3.5: Replikace XML dat pomocí ActiveSync

Jak vyplývá z obrázku, postačí v manuálním nebo poloautomatickém režimu synchronizovat soubory XML, které obsahují příslušná data.

Zápis datasetu do XML

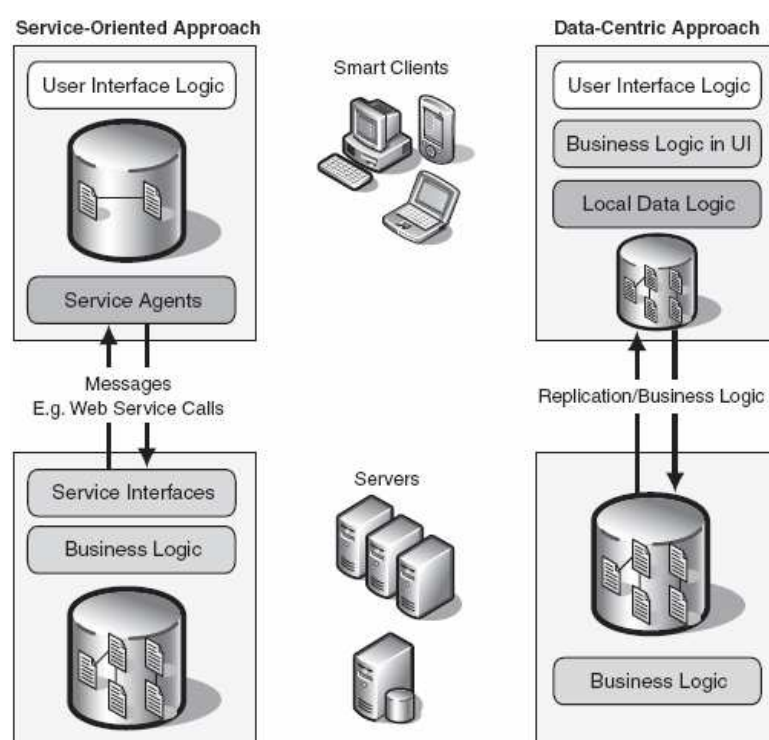
Pomocí nástrojů platformy .NET je synchronizace dat mezi odloučeným pracovištěm a datovou základnou ve firmě velice snadná. Při přenosu z PC na PDA stačí data z databáze uložit do DataSet a z DataSet do souboru XML. Soubor XML se poté lehce přenáší do mobilního zařízení. Mobilní zařízení potom s tímto XML souborem může pracovat jako s lokální datovou základnou.

3.2 Způsoby synchronizace dat

Většina uživatelů mobilních aplikací požaduje, aby aplikace, která pracuje v režimu online a komunikuje pomocí webových služeb nebo remotingu se serverem, byla schopna správně pracovat i při odpojení zařízení od sítě, tj. v režimu offline.

Existují dvě možnosti offline provozu aplikace po odpojení od sítě: tzv. přístup k centralizovaným datům a přístup k datům pomocí služeb.

Při přístupu k centralizovaným datům klient využívá lokální databáze a mechanismu merge replikací nabízených SQL serverem pro řízení dat v režimu offline. Naopak při synchronizaci s přístupem pomocí služeb klient spolupracuje s několika službami. Klient v offline režimu data podrží v cache do té doby, než bude znovu připojen k webové službě.



Obr. 3.7: Synchronizace zaměřená na přístup přes služby a přístup k centralizovaným datům, zdroj Microsoft MSDN

Přístup k centralizovaným datům

Při tomto druhu synchronizace se využívá mechanismu tzv. merge replikací. Klient nejprve vytvoří takzvanou subskripci na data, které bude potřebovat stáhnout z „velkého“ serveru do té doby, než bude odpojen od sítě. Jakmile klient je offline, aplikace pracuje s lokální databází. Když se klient zase zpátky připojí k síti, data jsou řízena mechanismem merge replikací a jsou aktualizována se serverem.

Přístup k datům pomocí služeb

V tomto případě se většinou využívá webových služeb. Klient a server jsou vzájemně nezávislí.

Aby klient byl schopný pracovat v odpojeném stavu, musíme na klientovi vytvořit infrastrukturu, která by umožnila uchovávat některé požadavky na služby tak, aby je klient mohl vykonat po připojení k síti. Tato infrastruktura se skládá ze 4 elementů:

- Service Agent. Zajišťuje připojení k webové službě. Zprostředkovává spolupráci všech klientů se službami a drží speciální logiku pro vykonávání žádostí klientů na služby.
- Žádost na službu. Všechny žádosti na služby drží tento objekt. Dále pak jsou požadavky odeslány do fronty žádostí a drží se v ní, pokud nebudou vykonány.
- Fronta žádostí. Všechny žádosti na služby jsou ukládány do fronty žádostí.
- Zpracovatel. Vybírá žádosti z fronty a vykoná je v momentě, kdy se klient připojí k síti.

Pro využití synchronizace v softwarové aplikaci musíme zajistit následující funkčnost aplikace

- Detekce připojení k síti
- Uchovávání dat v cache nebo lokální databázi pro zachování funkčnosti aplikace i při práci offline.
- Synchronizace dat mezi klientem a serverem po připojení aplikace k síti.



Obr. 3.6: Princip synchronizace, zdroj: Microsoft

3.3 Ukázková aplikace

První ukázková aplikace, kterou uvedeme v této diplomové práci, je z oblasti provozu restauračního zařízení. Nápad naprogramovat tuto aplikaci se zrodil během oběda v restauraci Nisa (nyní již bývalé). Jak je známo, je tam vždy přes poledne velký provoz a servírky často nestíhají. Proto jsem se rozhodla naprogramovat jednu ukázkovou aplikaci navíc, a to právě pro potřeby servírek v restauraci.

Níže bude uvedena částečná specifikace aplikace. V těle diplomové práce není příliš prostoru pro vložení kompletní specifikace aplikace a kompletního popisu řešení tak, jak je to obvyklé v praxi. Proto je kompletní specifikace uvedena v příloze A. Zde se podíváme pouze na nejdůležitější kroky.

Současné řešení

V současné době zákazník používá následující způsoby evidence informací:

Servírka zapisuje objednávku na papír, pak jde přes celou jídelní místnost do kuchyně a předává kuchařům objednávku. Jeden z kuchařů si tuto objednávku znovu zapisuje na papír a začíná s přípravou jídel. Informace o objednateli se ještě nějakou dobu uchovává na lístku servírky a po zaplacení zákazníkem se úplně ztrácí.

Tento „manuální“ systém neumožňuje zákazníkovi pracovat s daty dle jeho potřeb. Předávání dat mezi servírkou a kuchyní je pomalé. Servírka v čas oběda musí za co nejkratší dobu obsloužit co nejvíce zákazníků a zároveň včas předávat objednávky kuchařům. Ředitel restaurace by si přál nejen urychlit systém předávání objednávek mezi servírkami a kuchyní, ale také automaticky vypočítávat částku za objednávku a mít přehled o nejvíce a nejméně objednávaných jídlech.

Význam vyvíjené aplikace

Hlavní přínos aplikace „RestSoft“ spočívá v rychlém předávání informací mezi jídelnou a kuchyní. Mezi očekávanými pozitivními důsledky by mělo být:

- automatický výpočet částky za objednávku a tisk účtenky
- finanční přehled pro vedení restaurace
- přehled o nejvíce a nejméně objednávaných jídlech
- rozšiřitelný systém založený na nejmodernějších technologiích

Uživatelé systému

Cílovou skupinu uživatelů aplikace „RestSoft“ tvoří zaměstnanci restauračního zařízení. Pro role, které jednotliví uživatelé v systému zastávají, se v modelování užívá výraz aktéři.

Aktér	Funkce
Waitress	Servírka. Je to uživatel, který má u sebe mobilní přístroj a ukládá do systému nové objednávky. Dále pak tiskne účtenky ze systému.
Cook	Kuchař. Uživatel, který může přistupovat k datům všech aktuálních objednávek a označovat ty z nich, které už jsou uvařeny .
Management	Vedení. Uživatel, který má přístup do všech částí aplikace. Má přehled o všech objednávkách.
Administrator	Administrátor. Má přístup do všech částí aplikace. Může měnit nastavení aplikace i nastavení databáze.

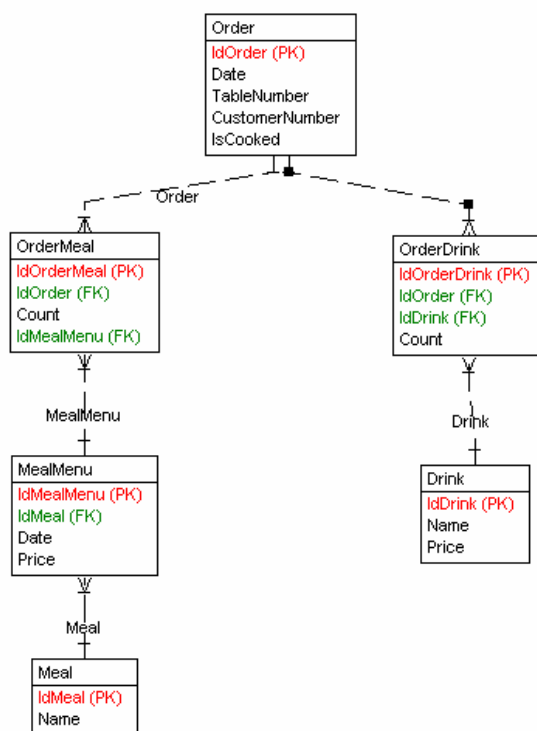
Tab. 3.7: Role uživatelů ve vztahu k „RestSoft“ aplikaci

Popis řešení

Koncepce výsledného řešení vychází z následujících předpokladů:

- systém je vybudován za užití objektových postupů návrhu vícevrstvé architektury, kde je obchodní logika aplikace prováděna odděleně od prezentační části aplikace
- všechna data jsou ukládána v relační databázi na serveru
- uživatelské rozhraní na serveru (počítač na pokladně a v kuchyni) je vytvořeno jako aplikace běžící pod MS Windows (tlustý klient)
- uživatelské rozhraní na klientu (mobilní zařízení v rukou servírek) je vytvořeno jako aplikace pro mobilní aplikace (tlustý klient)
- data mezi klientem a serverem jsou přenášena za využití webových služeb
- mobilní zařízení mohou být připojena k síti přes Wi-Fi, bluetooth nebo zařízení bude připojeno k internetu.

Propojení dat v systému



Obr. 3.8: Databázová strukturu IS RestSoft

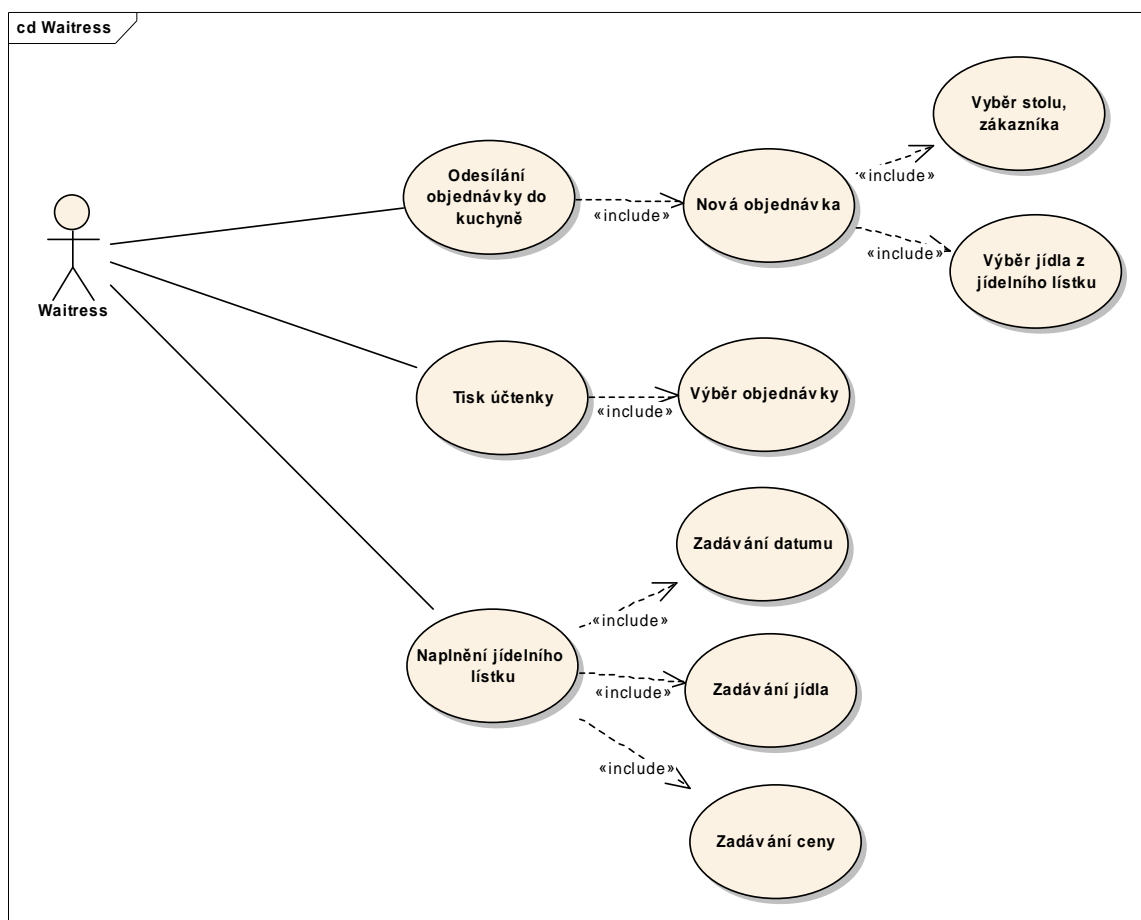
Výstupy

Standardními výstupy jsou:

- tiskové sestavy všech významných dat
- export dat do souborů aplikace MS Excel

Popis užití

Činnost servírky



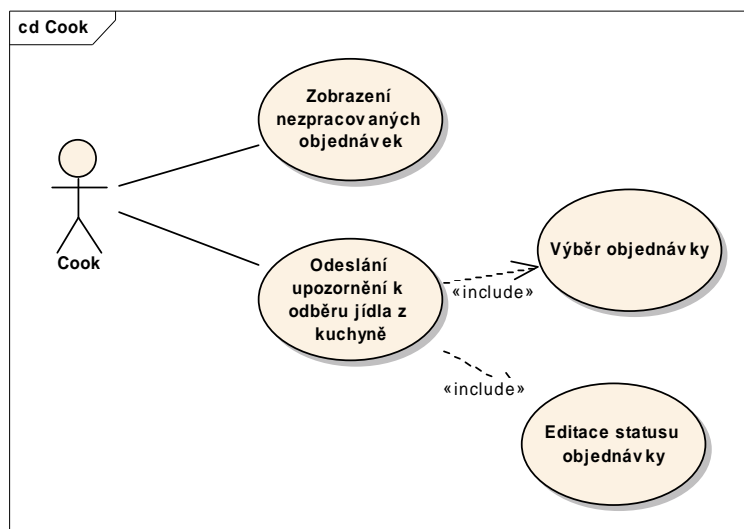
Obr. 3.9: UseCase diagram Činnost servírky

Jednou z hlavních částí systému je práce se zákaznickou objednávkou. Z pohledu celého systému se pod pojmem Zákazník rozumí právnická či fyzická osoba, která přijde do restaurace na jídlo. Pod pojmem Objednávka se rozumí zákazníkem objednané jídlo a pití. Výše zobrazený diagram popisuje základní členění práce s objednávkou.

Servírka nejdřív přijde obsloužit zákazníka. Do informačního systému v PDA založí novou zákaznickou objednávku. Dále musí zvolit číslo stolu a číslo zákazníka (u jednoho stolu mohou sedět cizí lidi, např. v restauraci typu bývala Nisa, Liberec). Výběrem jídla z denního jídelního lístku a nápoje z nápojového lístku může objednávku uložit a odeslat do kuchyně, kde se začne jídlo připravovat.

Další funkcí systému by měl být tisk účtenky, kdy servírka přímo z PDA může vytisknout účtenky pro zvolenou objednávku.

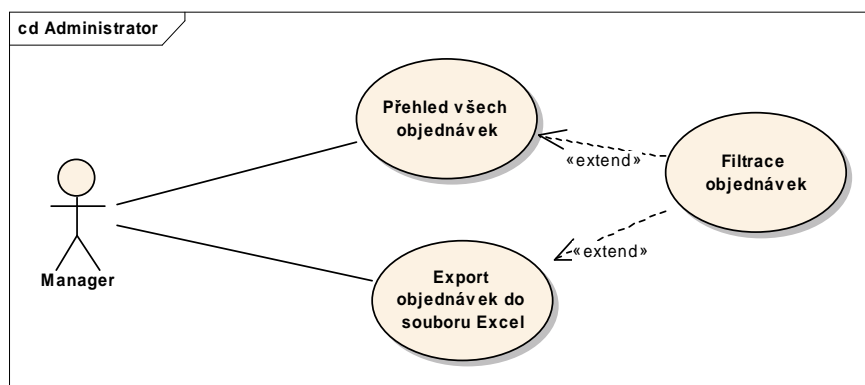
Činnosti kuchařů



Obr. 3.10: UseCase diagram Činnost kuchaře

Kuchařovi se na displeji počítače v kuchyni objeví aktuální seznam nezpracovaných objednávek. Poté co je objednávka připravená, odešle kuchař upozornění servírce (na příslušné PDA), od níž objednávku obdržel.

Činnost administrátora (manažera restaurace)



Obr. 3.11: UseCase diagram Činnost administrátora

Administrátor, resp. manažer restauračního zařízení, musí mít přehled o všech objednávkách za určité období a také možnost exportovat potřebná data do souboru Excel. Podrobnější popis systému s ukázkami programu je uveden v příloze 1.

4 Použití IT technologií v oblasti stavebnictví

V zadání diplomové práce je uvedeno zpracování ukázkové aplikace z oblasti stavebnictví, v této kapitole se tedy podíváme na využití mobilních zařízení ve stavebnictví.

4.1 Test mobilních zařízení (PDA) na stavbě

V oblasti stavebnictví existuje několik různých pracovních náplní, která zahrnují různá řemesla a proto různé typy lidí. To také odráží různou úroveň počítačové gramotnosti a různé vnímání aplikovatelnosti použití IT v jejich oboru.

Byly provedeny testy na pěti různých skupinkách lidí z různých oborů stavebnictví s cílem pochopit, jak různá zařízení vyhovují různým lidem a zda-li jim pomáhají v plnění jejich úkolů. 5 skupin pracovníků stavebnictví bylo zvoleno takto:

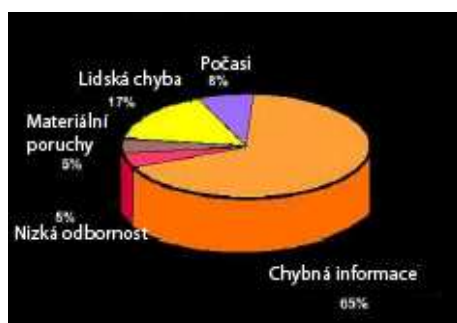
- Jednatel
- Stavbyvedoucí
- Stavební technik
- Správa stavby/Mistr
- Poradce pro zdraví, bezpečnost a ochranu prostředí

Zařízení se testovala v každodenním stavebním provozu při plnění pracovních úkolů a také se testovalo, zda-li jsou tato zařízení vhodná pro stavební sféru. Testovala se schopnost zařízení plnit hlavně tyto úkoly:

- prohlížení a kreslení na PocketCAD
- zaznamenávání výkresu pomocí PocketCAD
- čtení dokumentace
- zápis do diáře
- vyplnění formulářů

Další problém – práce s informacemi.

Obrázek 4.1 ukazuje, že 65% stavebních oprav je způsobeno nedostatečnými, nevhodnými nebo protichůdnými informacemi.



Obr. 4.1: Graf stavebních oprav, zdroj: The appropriate use of I.T. on a construction site

Většina projektových informací se ukládá na papír, kde je těžko přístupná a zabírá velkou kapacitu paměti. Automatizované informační systémy mohou efektivně skladovat ohromné množství dat a informace může být rychle umístěna a prohlížena pomocí automatizovaného vyhledávání a zobrazení.

4.2 Komunikace ve stavebnictví

Je potřeba, aby lidé, jenž pracují ve stavebnictví, byli mobilní. S rozvojem mobilních telefonů a bezdrátové komunikace existuje více a více možností ke zefektivnění komunikace. Cílem této diplomové práce je zjednodušení komunikace mezi zařízeními na dálku. Stavbyvedoucí je důležitou součástí každé stavby. Je hlavním stavebním pilířem stavby. Řídí celou stavbu, musí mít přehled o všem, co se děje; to znamená, že musí mít neustalý přístup k aktuálním informacím. Mobilní telefon je určitě dobrou pomůckou k přenosu dat. Ve většině případů ale umožňuje přenášet pouze hlasovou informaci. Pro přenos dat potřebuje mobilní přístroj, ať již to bude notebook nebo jiné mobilní zařízení, na které se nainstaluje jednoduchá aplikace, která bude komunikovat v režimu online nebo offline. Při použití notebooku můžeme využít .NET remotingu pro komunikaci se serverem (Compact Framework remoting nepodporuje). Při použití jiného mobilního zařízení, například PDA, můžeme využít webových služeb. Jiným řešením je lokální ukládání dat a následná synchronizace mezi klientem a serverem.

4.3 Software ve stavebnictví

Stavebnictví vyžaduje mobilitu. Komunikace s kolegy je podstatná a musí být kvalitní. Množství a načasování informací může buď bránit nebo naopak podporovat dosažení dobrých výsledků práce.

Mnozí výrobci softwaru tyto vazby pochopili správně a realizují konstrukční mobilní softwarová řešení pro oblast stavebnictví.

A jaké typy softwarových aplikací by ve stavebnictví byly využívány?

CAD Applications

CAD je nyní používán ve většině stavebních projektů ke kreslení výkresů. Nicméně, ačkoli jsou výkresy zhotoveny elektronicky, pro jejich další použití je potřeba jejich vytisknutí na papír. Tento fakt eliminuje mnohé výhody elektronického zpracování a snižuje efektivitu zpětné vazby.

Typické použití:

- Kreslení CAD výkresů na místě
- Výkresy skutečného stavu stavby
- Vypočítávání vzdáleností
- Přidání komentářů / poznámek



Obr. 4.2: CAD aplikace, zdroj: The appropriate use of I.T. on a construction site

Software pro zlepšení spolupráce

Nyní se vede hodně diskuzí o druhu softwaru, který řeší problémy rozdělení práce.

Typické použití:

- Poptávka po informacích
- Kontaktní údaje
- Výměna informací, např. výkresů

Software pro ukládání dat

Typické použití

- Na stavbě se do mobilního zařízení pomocí speciálního software vloží vyplněné formuláře nebo tabulky.
- Aktuální informace se buď prostřednictvím internetu odesílají hned na databázový server stavební firmy (online synchronizace), nebo se ukládají do mobilní databáze a poté se vykonává tzv. offline synchronizace, kdy mobilní přístroj je připojen k databázovému serveru přes kabel, wi-fi nebo přes adaptér bluetooth a data se tak synchronizují.



The image shows a software interface for data entry on a construction site. It features a series of dropdown menus for hierarchical location selection: Site (Century Wharf), Block (Block 1), Section (Ground Floor), Unit (A1 Apartment 78), and Room (Bathroom). Below these is a table with columns for Type, Snag, and Location. The table contains one entry: Walls, Unaven, E. To the right of the table are filters for Snag Filter (Walls) and Status Filter (Cleared). At the bottom are four buttons: New, Edit, Status, and Sync. A handheld PDA device is overlaid on the right side of the interface, displaying a similar data entry screen.

Site	Century Wharf
Block	Block 1
Section	Ground Floor
Unit	A1 Apartment 78
Room	Bathroom

Type	Snag	Loc
Walls	Unaven	E

Snag Filter: Walls
Status Filter: Cleared

Buttons: New, Edit, Status, Sync

Obr. 4.3: Software pro ukládání dat, zdroj: The appropriate use of I.T. on a construction site

5 Praktická aplikace v podmínkách stavební firmy

– stavbyvedoucí a sídlo firmy

Další ukázková aplikace, kterou uvedeme v této diplomové práci, je z oblasti stavebnictví. Jak již bylo uvedeno výše, činnost stavbyvedoucího na stavbě je velmi důležitá. Neméně důležitá je dobrá komunikace se sídlem firmy. Proto bylo navrženo zadání této práce.

Níže uvedeme částečnou specifikaci aplikace, analýzu řešení a ukázky z této aplikace. Kompletní specifikace je uvedena v příloze B. Zde uvedeme pouze nejdůležitější kroky.

Popis současného stavu

V současné době zákazník (stavbyvedoucí) používá následující způsoby evidence informací. Stavbyvedoucí zapisuje všechny informace na papír a poté je pracně přepisuje do počítače v kanceláři. Největším problémem je dvojitá práce a také ten fakt, že data na firemním serveru nejsou aktuální, stavbyvedoucí na přepisování informací z papíru většinou nemá čas a dodává je do systému na poslední chvíli na konci období.

Informace zapsané na papír obsahují spoustu chyb, jelikož takový systém nemá žádný řád. Také se často stává, že se data ztratí úplně nebo i jen částečně.

Na druhé straně stavbyvedoucí v některých případech potřebuje přehled o aktuálních datech. Například při vedení stavové evidence potřebuje mít přehled o aktuálním stavu materiálu na skladu (stavbě).

Význam vyvíjené aplikace

Hlavní přínos aplikace „BuildingSoft“ spočívá v usnadnění a zpřehlednění práce stavbyvedoucího, zrychlení předávání informací mezi stavbou a kanceláří, zmenšení papírování, automatizace práce. Mezi očekávanými pozitivními důsledky by mělo být:

- evidence docházky pracovníků na stavbě a následný výpočet mezd v účtárně.
- stavební evidence – evidence stavu materiálu na skladě (stavbě)
- evidence čerpání položek rozpočtu
- synchronizace dat mezi klientem a serverem

Popis uživatelů

Tato podkapitola popisuje rozdělení uživatelů, definuje jejich role a typy podle jejich vztahu k informačnímu systému. Cílovou skupinu uživatelů aplikace „BuildingSoft“ tvoří stavbyvedoucí na stavbě.

Aktér	Funkce
Stavbyvedoucí	Je to uživatel, který má u sebe přístroj (mobilní zařízení PDA) nebo notebook, ukládá do systému docházku pracovníků, stavební evidenci a čerpá rozpočet stavby. Díky aktuálním informacím má přehled o stavu stavby, zboží a docházky. Synchronizuje data ze svého přístroje s databází na serveru ve firmě, tím udržuje serverovou databázi aktuální, nebo v případě čerpání se data po připojení synchronizují sama.

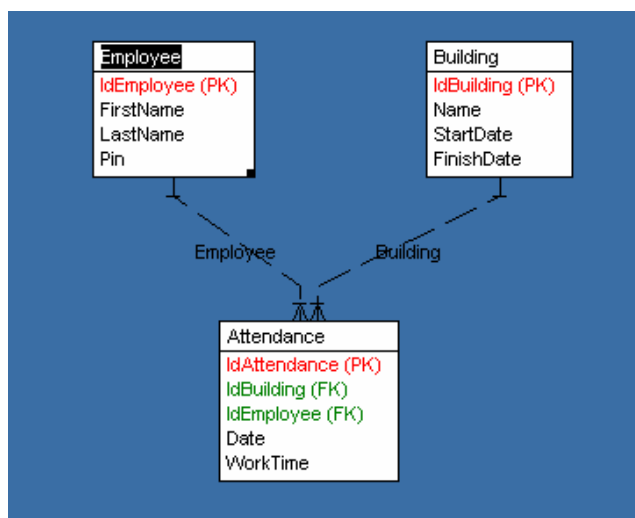
Tab. 5.1: Role uživatelů ve vztahu k „BuildingSoft“

Řešení

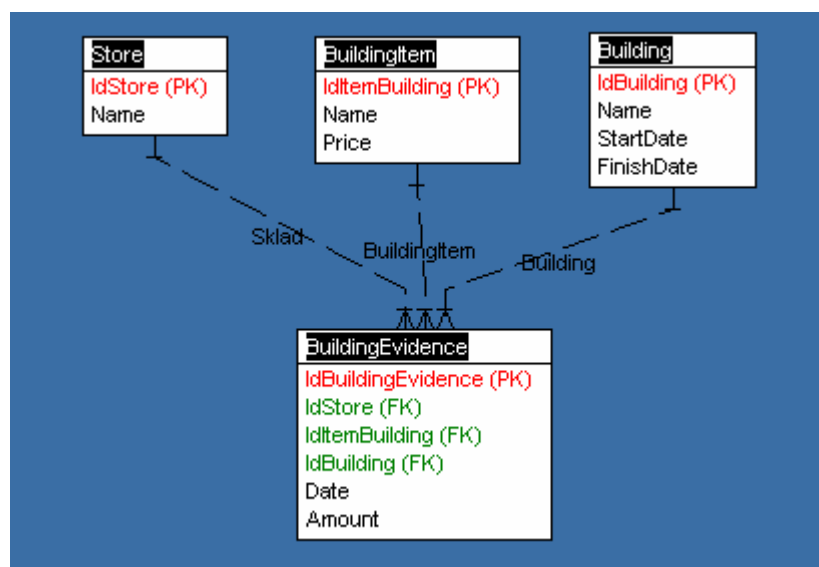
Koncepce výsledného řešení vychází z následujících předpokladů:

- systém je vybudován za užití objektových postupů návrhu vícevrstvé architektury, kde je obchodní logika aplikace prováděna odděleně od prezentační části aplikace
- všechna data jsou ukládána v relační databázi
- databáze na serveru (serverový počítač v kanceláři) je vytvořena jen pro ukázkou synchronizace. IS se nezabývá funkčností serverové aplikace
- uživatelské rozhraní na klientu (mobilní zařízení) je vytvořeno jako mobilní aplikace (tlustý klient)
- mobilní databáze na klientu je vytvořena pouze pro lokální přístup z klienta a po připojení k počítači v kanceláři bude možné synchronizovat tuto databázi.
- pro čerpání stavebního rozpočtu je použit Offline aplikační blok, který zajišťuje funkčnost aplikace v režimu offline.

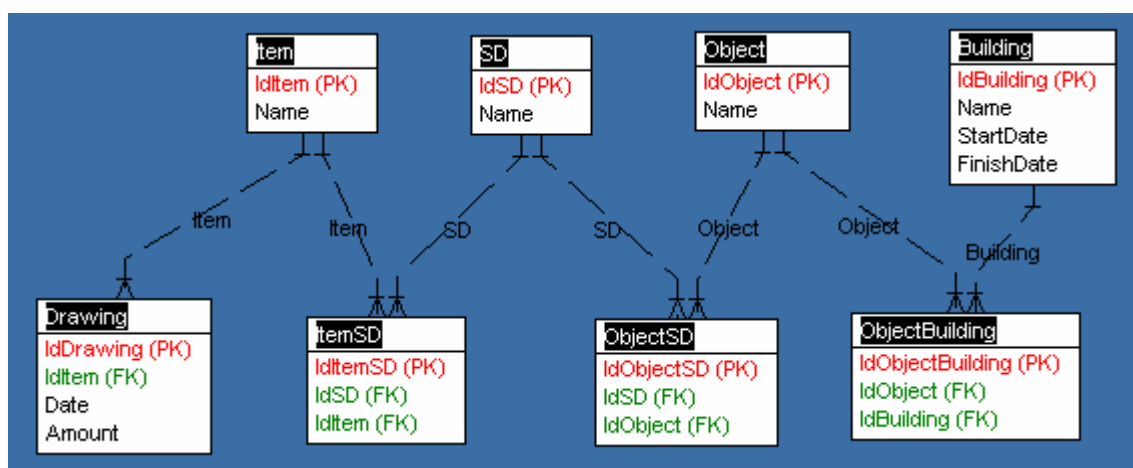
Propojení dat v systému



Obr.5.2: Struktura databáze pro zadávání docházky



Obr.5.3: Struktura databáze pro vedení stavební evidence



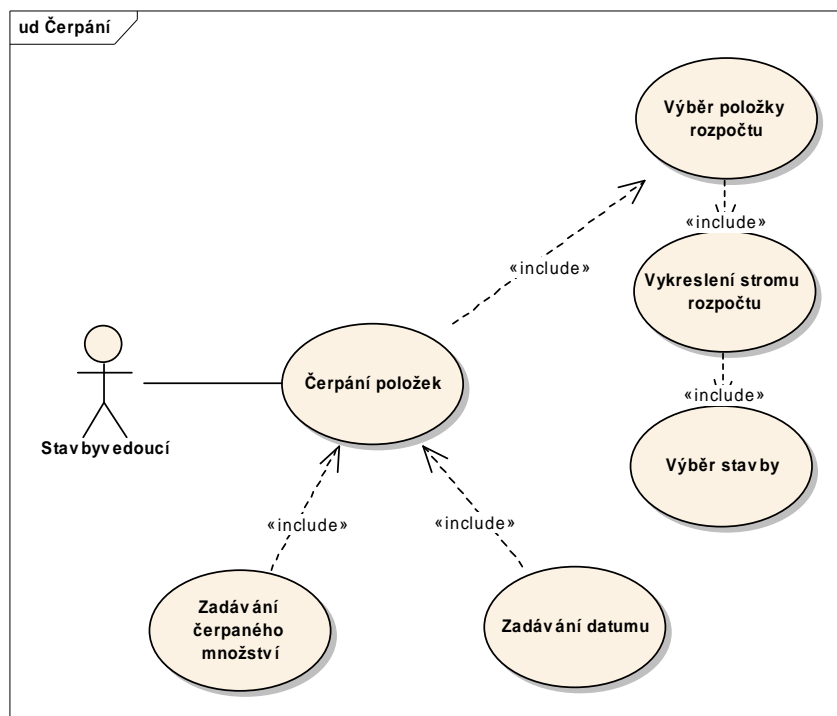
Obr.5.4: Struktura databáze pro čerpání položek rozpočtu

Popis užití

Čerpání rozpočtu

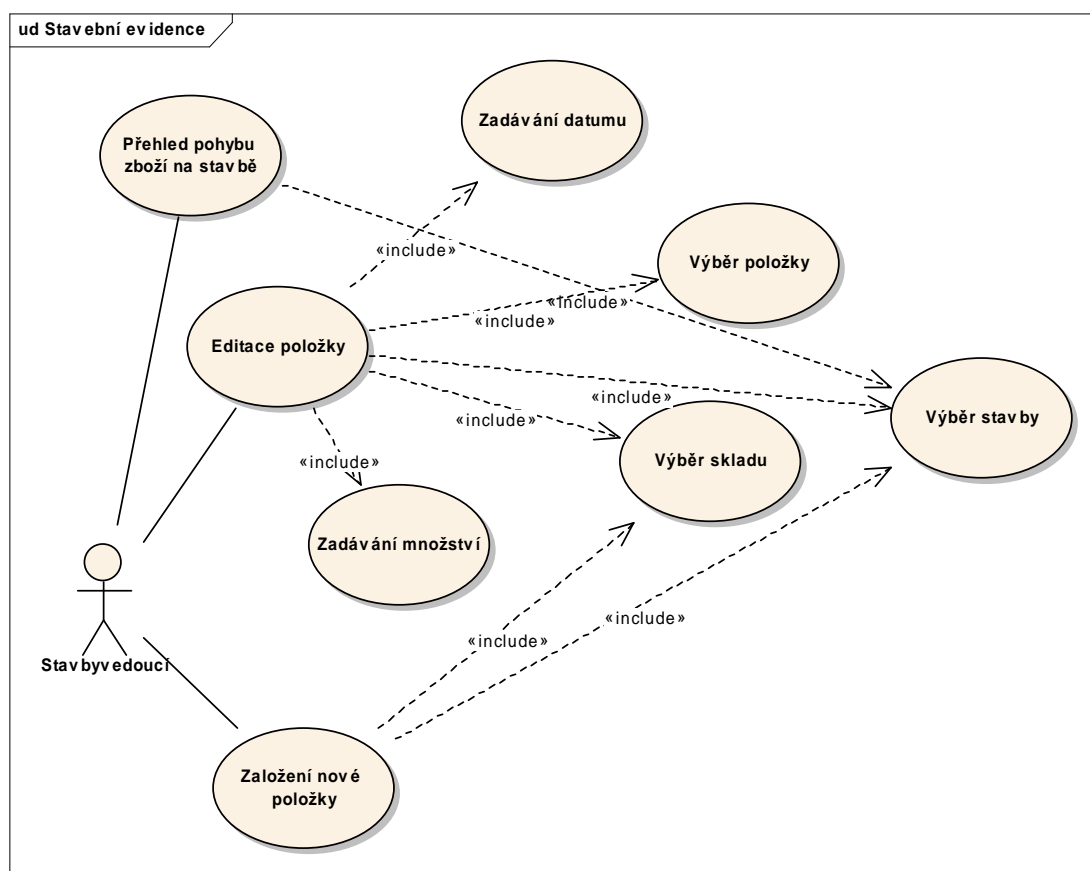
Čerpání položek stavebního rozpočtu bude prováděno v režimu offline i online současně. To znamená, že stavbyvedoucí nemusí být vždy připojen k síti. Aplikace sama zjistí stav připojení a pracuje s daty buď z cache v případě offline režimu nebo přímo ze serverové databáze (online režim).

Přenos dat může být proveden buď pomocí webových služeb nebo pomocí remotingu (pokud stavbyvedoucí používá pro práci notebook).



Obr. 5.5: Use Case diagram: čerpání položek rozpočtu

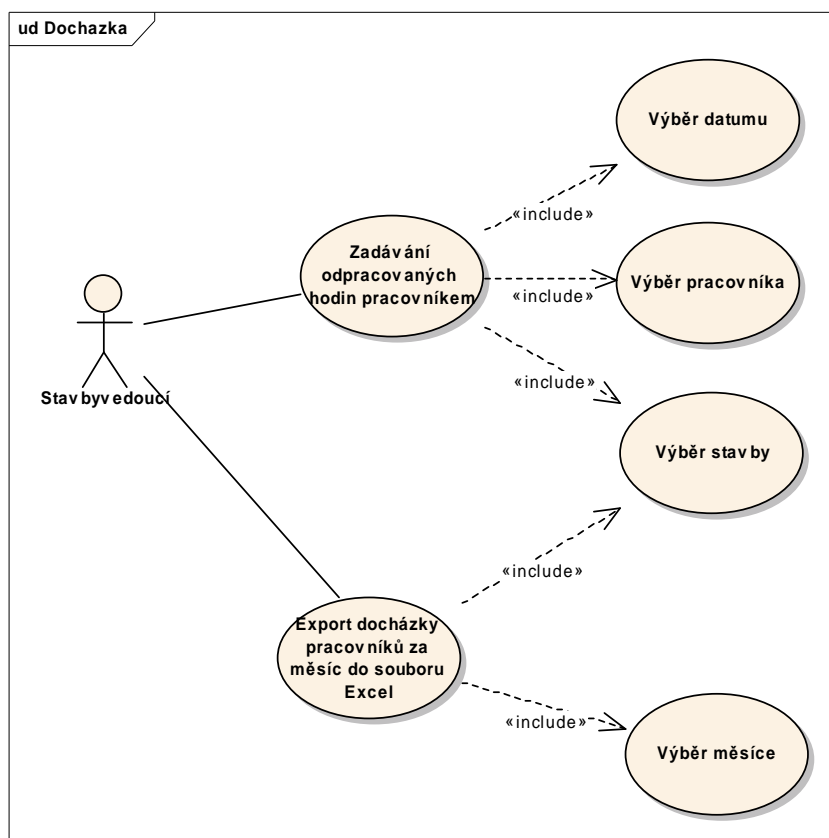
Stavební evidence



Obr. 5.6: Use Case diagram: stavební evidence

V této aplikaci může stavbyvedoucí zakládat nové stavební položky, editovat čerpané množství položek a přidávat množství do evidence. Přímou ve svém PDA si stavbyvedoucí může prohlédnout, jaké množství jakých položek má k dispozici na kterém skladu.

Docházka zaměstnanců na stavbě



Obr. 5.7: Use case diagram: Docházka zaměstnanců

Stavbyvedoucí může zadávat a mít přehled o docházce zaměstnanců na stavbě a pak tuto informaci například exportovat do souboru Excel a předat účetní.

Podrobnější popis aplikace s ukázkami je uveden v příloze B.

Závěr

V rámci diplomové práce bylo nalezeno optimální řešení pro sdílení dat na dálku. Byla prozkoumána problematika bezdrátové výměny informací, současná řešení a jako hlavní výsledek bylo vypracováno několik jednoduchých softwarových aplikací pro bezdrátovou komunikaci mezi klientem a serverem, které jsou univerzální a umožňují rychle a efektivně posílat důležitá data na pracovní server.

1) *Technologická část:* Byly zmapovány jednotlivé technologické možnosti pro bezdrátovou komunikaci. Těchto možností je několik – od mobilních telefonů po satelitní technologie. Některé z nich ale zaručují hlasový avšak ne datový přenos. V rámci této práce byla vybrána jen taková řešení, která by poskytovala hlavně možnost datového přenosu.

2) *Způsob přenosu dat:* Jedním z hlavních úkolů této práce bylo nalezení takového způsobu přenosu dat na dálku, jenž by zaručoval jednoduchost, rychlost, bezpečnost a zejména univerzálnost. Byl vyzkoušen online a offline režim přenosu dat. Při online režimu (např. v aplikaci RestSoft) jsou data ihned odesílána na server a přijímána ze serveru. Offline režim (např. v aplikaci BuildingSoft) dovoluje uchovávat data na klientovi, ať už v lokální databázi nebo v cache a po připojení k síti jsou data odeslána na server.

3) *Softwarové řešení:* Hlavním úkolem bylo vypracovat takovou softwarovou aplikaci, pomocí které by bylo umožněno sdílení informací na dálku. Nakonec se podařilo vypracovat několik aplikací, které ukazují jednotlivá možná řešení přenosu dat. Architektura všech aplikací byla navržena klasicky: klient – server. Jako klient se rozumí mobilní zařízení schopné pracovat s datovým přenosem – v tomto případě PDA. Jako server se rozumí klasický databázový server běžící v sídle firmy. Aplikace nabízejí formuláře pro zadávání důležitých dat na klientu PDA, navazují spojení se serverem a aktualizují data.

Důležitým a pozitivním výsledkem této práce je její univerzálnost. Konečné aplikace sice neobsahují celou funkčnost zpracovávané problematiky, není ale složité tyto aplikace rozšířit a využívat nejenom v oblasti stavebnictví a restauračních zařízení. Jednotlivá řešení, zpracovávaná v této práci, mohou sloužit jako stavební kameny k další činnosti.

Seznam použité literatury

- [1] *Wikipedia, the free encyclopedia*. [online]. [cit. 2006-04-02]
URL: <<http://en.wikipedia.org>>
- [2] *Computerworld, internetový portál o I.T. technologiích* [online].
[cit. 2006-09-02]
URL: <<http://www.computerworld.cz/>>
- [3] *Microsoft MSDN, báze znalostí o platformě .NET* [online].
[cit. 2006-09-01]
URL: <<http://www.msdn2.microsoft.com/>>
- [4] Tomáš Wyszczor: *.NET remoting pod lupou*, 2004
- [5] ARUP: *Appropriate use of I.T. on a construction site*, 2005
- [6] Microsoft: *Offline Application Block Help*, 2004
- [7] Joseph Schmuller: *Mysl9me v jazyku UML*, Grada, 2001
- [8] Imrich Buranský: *XML a webové služby*, Microsoft, 2004
- [9] Ľuboslav Lacko: *Vývoj aplikácií pre mobilné zariadenia*, Microsoft, 2004
- [10] Ľuboslav Lacko: *Vývoj webových aplikácií pre začiatočníkov pomocou Express nástrojov*, Microsoft, 2003

	Název dokumentu: Specifikace IS RestSoft pro libovolné restaurační zařízení
--	---

Zpracoval:	Oxana Shamanina
Verze dokumentu:	
Datum vytvoření:	9.9.2006
Počet stran:	15

Revize:

Datum	Verze	Popis	Autor
9.9.2006	1.0	Vytvoření dokumentu	Oxana Shamanina

Obsah – Příloha A

Obsah – Příloha A	55
Úvod.....	56
Účel.....	56
1 Oblasti k řešení	56
1.1 Popis současného řešení.....	56
1.2 Význam vyvíjené aplikace.....	56
2 Popis okolí systému	57
2.1 Popis uživatelů.....	57
3 Charakteristika produktu	58
3.1 Koncepce řešení.....	58
3.2 Vlastnosti systému	59
3.2.1 Popis užití a ukázky z aplikace	59
3.2.2 Propojení dat v systému.....	65
3.2.3 Výstupy.....	66
3.2.4 Diagramy tříd.....	66
4 Požadavky.....	68
4.1 Systémové požadavky.....	68
4.2 Výkonové požadavky	68

Úvod

Cílem tohoto dokumentu je shromáždění, analýza a definice požadavků a vlastností systému „Objednávání jídel v restauračním zařízení“ dále jen „Rest“ včetně navrhovaných variant řešení. Dokument se zaměřuje na definici a zdůvodnění potřeb zákazníka a cílových uživatelů.

Účel

Účelem tohoto dokumentu je definovat požadavky na nejvyšší úrovni abstrakce a tedy určit rozsah vyvíjené aplikace. Informace jsou podány tak, aby byly zcela srozumitelné pro zákazníka.

1 Oblasti k řešení

1.1 Popis současného řešení

V současné době zákazník používá následující způsoby evidence informací:

Servírka zapisuje objednávku na papír, pak jde přes celou jídelní místnost do kuchyně a předává kuchařům objednávku. Jeden z kuchařů si tuto objednávku znovu zapisuje na papír a začíná s přípravou jídel. Informace o objednavce se ještě nějakou dobu uchovává na lístku servírky a po zaplacení zákazníkem se úplně ztrácí.

Tento „manuální“ systém neumožňuje zákazníkovi pracovat s daty dle jeho potřeb. Předávání dat mezi servírkou a kuchyní je pomalé. Servírka v čase oběda musí za co nejkratší dobu obsloužit co nejvíce zákazníků a zároveň včas předávat objednávky kuchařům. Ředitel restaurace by si přál nejen urychlit systém předávání objednávek mezi servírkami a kuchyní, ale také automaticky vypočítávat částku za objednávku a mít přehled o nejvíce a nejméně objednávaných jídlech.

1.2 Význam vyvíjené aplikace

Hlavní přínos aplikace „RestSoft“ spočívá v rychlém předávání informací mezi jídelnou a kuchyní. Mezi očekávanými pozitivními důsledky by měly být:

- automatický výpočet částky za objednávku a tisk účtenky
- finanční přehled pro vedení restaurace
- přehled o nejvíce a nejméně objednávaných jídlech
- rozšiřitelný systém založený na nejmodernějších technologiích

2 Popis okolí systému

Tato kapitola popisuje okolí a hranice systému – to znamená uživatele a jejich rozdělení podle jejich vztahu k vyvíjenému informačnímu systému. Dále popisuje strukturu a význam uživatelských rolí k jednotlivým entitám aplikace.

2.1. Popis uživatelů

Tato kapitola popisuje rozdělení uživatelů, definuje jejich role a typy podle jejich vztahu k informačnímu systému. Cílovou skupinu uživatelů aplikace „RestSoft“ tvoří zaměstnanci restauračního zařízení.

Pro role, které jednotliví uživatelé v systému zastávají, se v modelování užívá výraz aktéři.

Aktér	Funkce
Waitress	Servírka. Je to uživatel, který má u sebe přístroj vybavený zařízením Bluetooth a ukládá do systému nové objednávky. Dále pak tiskne účtenky ze systému.
Cook	Kuchař. Uživatel, který může přistupovat k datům všech aktuálních objednávek a označovat ty z nich, které už jsou uvařeny .
Management	Vedení. Uživatel, který má přístup do všech částí aplikace. Má přehled o všech objednávkách.
Administrator	Administrátor. Má přístup do všech částí aplikace. Může měnit nastavení aplikace i nastavení databáze.

Tab. 1: Role uživatelů ve vztahu k „Rest“

3 Charakteristika produktu

3.1 Koncepce řešení

Koncepce výsledného řešení vychází z následujících předpokladů:

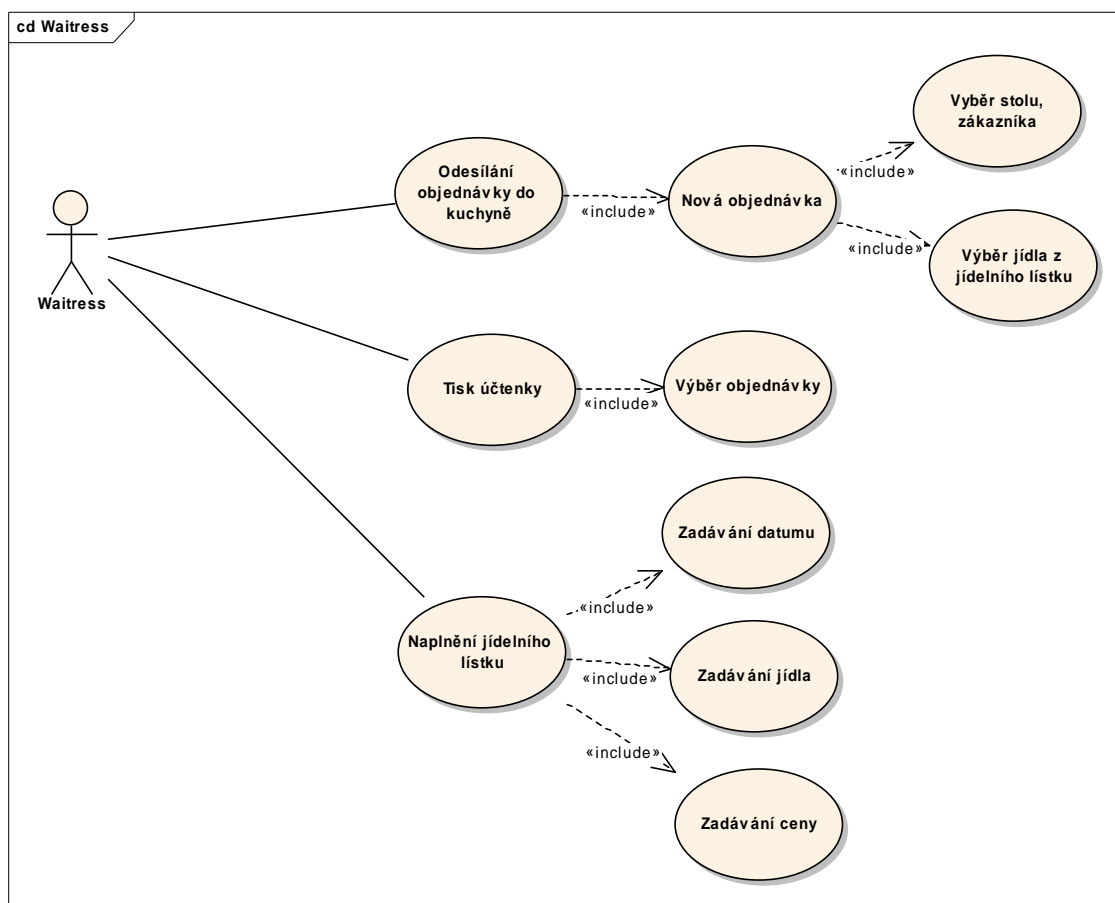
- systém bude vybudován za užití objektových postupů návrhu vícevrstvé architektury, kde je obchodní logika aplikace prováděna odděleně od prezentační části aplikace
- všechna data jsou ukládána v relační databázi
- uživatelské rozhraní na serveru (počítač na pokladně a v kuchyni) bude vytvořeno jako aplikace běžící pod MS Windows (tlustý klient)
- uživatelské rozhraní na klientu (mobilní zařízení v rukou servírek) bude vytvořeno jako aplikace pro mobilní aplikace (tlustý klient)

Návrh architektury

Hlavní princip návrhu je založen na propojení klienta (PDA) se serverem. Tento problém je vyřešen pomocí webových služeb. Zařízení PDA bude mít v sobě pouze vizuální část aplikace a knihovnu pro volání webových služeb. Server v sobě drží celou logiku IS.

3.2 Vlastnosti systému

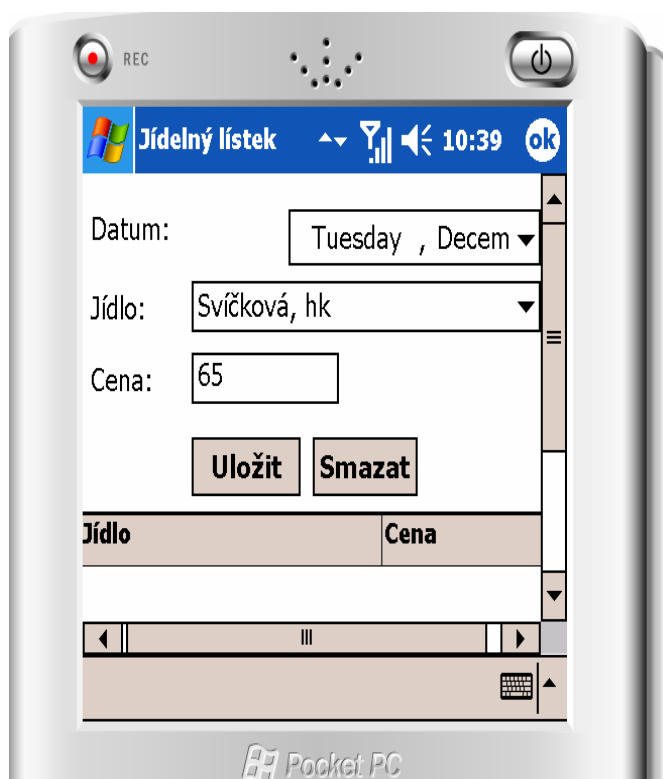
3.2.1 Popis užití a ukázky z aplikace



Obr. 1: UseCase diagram Činnost servírky

Jednou z hlavních částí systému je práce se zákaznickou objednávkou. Z pohledu celého systému se pod pojmem Zákazník rozumí právnická či fyzická osoba, která přijde do restaurace na jídlo. Pod pojmem Objednávka se rozumí zákazníkem objednávané jídlo a pítí. Výše zobrazený diagram popisuje základní členění práce s objednávkou.

Servírka nejdříve přijde obsloužit zákazníka. Do informačního systému v PDA založí novou zákaznickou objednávku. Dále musí zvolit číslo stolu a číslo zákazníka (u jednoho stolu můžou sedět cizí lidi, např. v restauraci typu bývalá Nisa, Liberec). Výběrem jídla z denního jídelního lístku a nápoje z nápojového lístku může objednávku uložit a odeslat do kuchyně, kde se začne jídlo připravovat.



Obr. 2: Ukázka aplikace – zadávání jídelního lístku

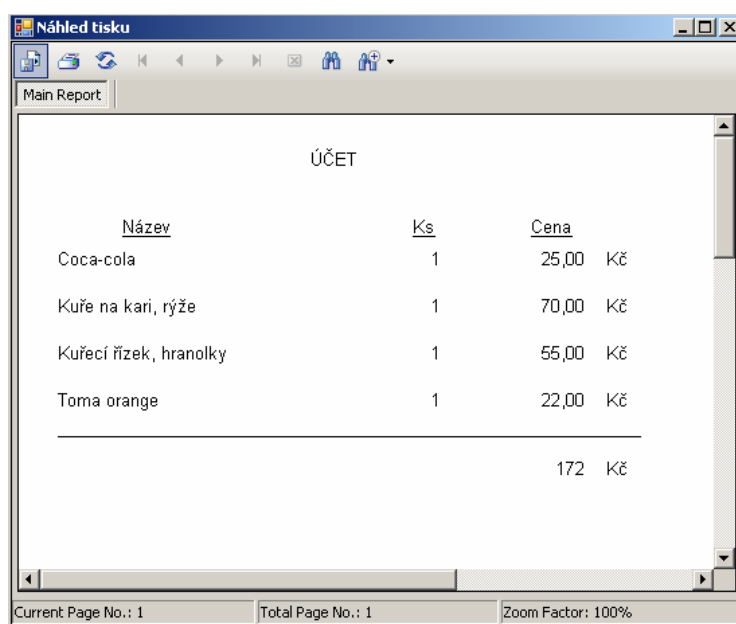


Obr. 3: Ukázka aplikace – výběr nápojů do nápojového lístku



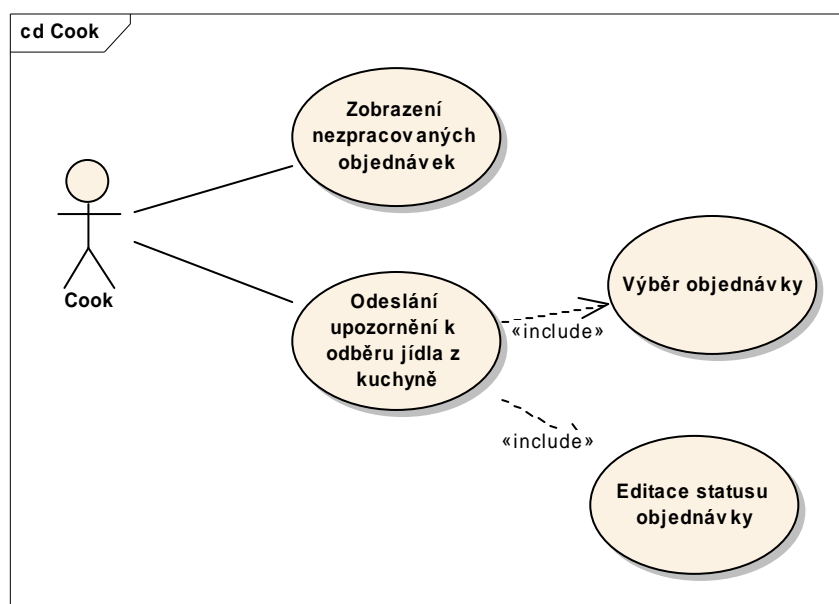
Obr. 4: Ukázka aplikace: Odeslání objednávky do kuchyně

Další funkcí systému by měl být tisk účtenky, kdy servírka přímo z PDA může vytisknout účtenku pro zvolenou objednávku.



Obr. 5: Ukázka aplikace: Tisk účtenky

Činnosti kuchařů



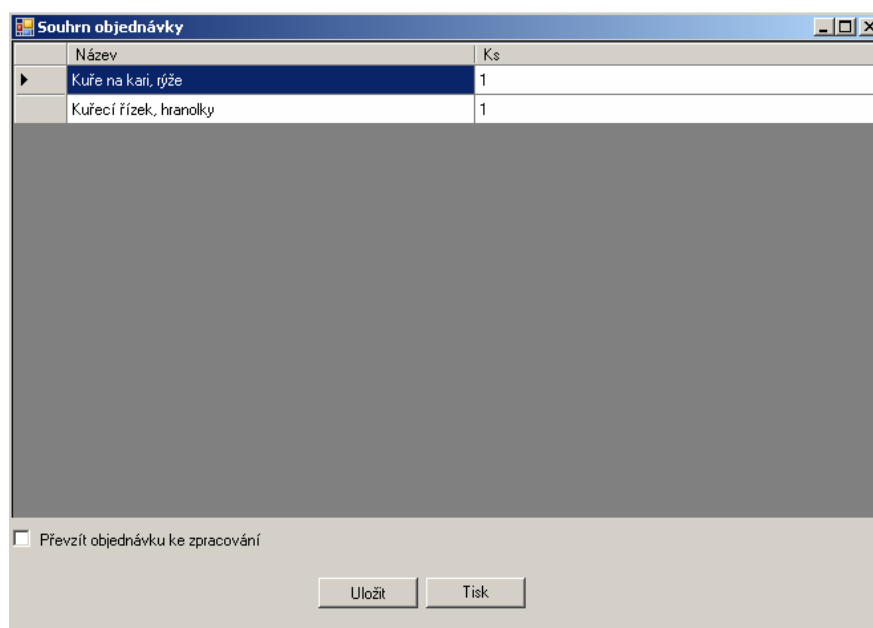
Obr. 6: UseCase diagram Činnost kuchaře

Objednávky				
Menu				
	Číslo zákazníka	Číslo stolu	Datum	Stav
▶	1	1	2.12.2006	<input type="checkbox"/>
	1	2	2.12.2006	<input type="checkbox"/>

Obr. 7: Ukázka aplikace: Počítač v kuchyni. Přehled všech nezpracovaných objednávek

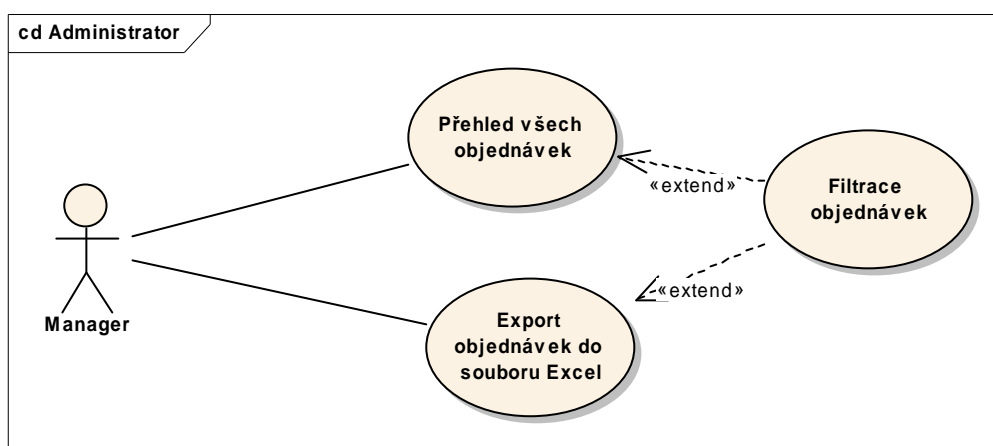
Kuchařovi se na displeji počítače v kuchyni objeví aktuální seznam nezpracovaných objednávek. Poté, co je objednávka připravená, odešle kuchař upozornění servírce (na příslušné PDA), od níž objednávku obdržel.

Po kliknutí na objednávku vidíme přehled objednaných jídel a nápojů a možnost změnit stav objednávky z nepřípravené na připravenou.



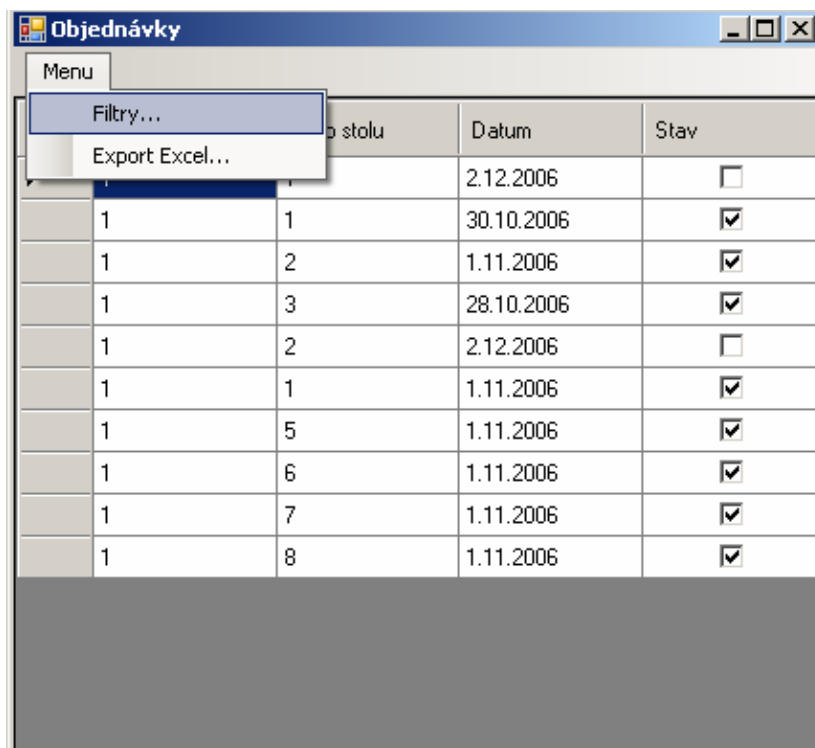
Obr. 8: Přehled objednávky

Činnost administrátora (manažera restaurace)



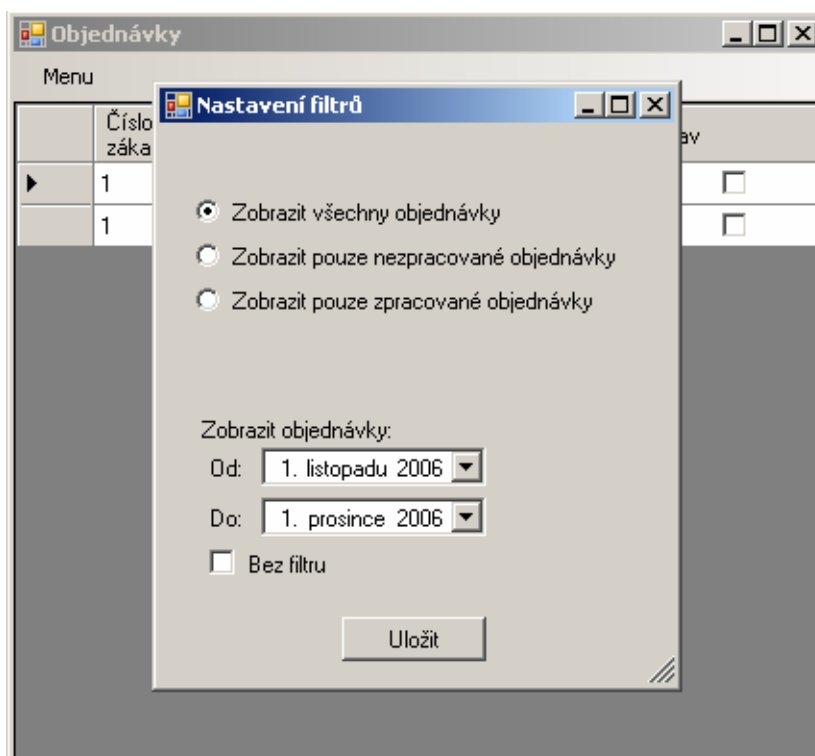
Obr. 9: UseCase diagram: činnost administrátora

Manažer restaurace si může nechat zobrazit přehled všech objednávek za určité období.



Obr. 9: Ukázka aplikace: Přehled všech vyfiltrovaných objednávek

Nastavit filtry je možné pomocí speciální funkce aplikace.



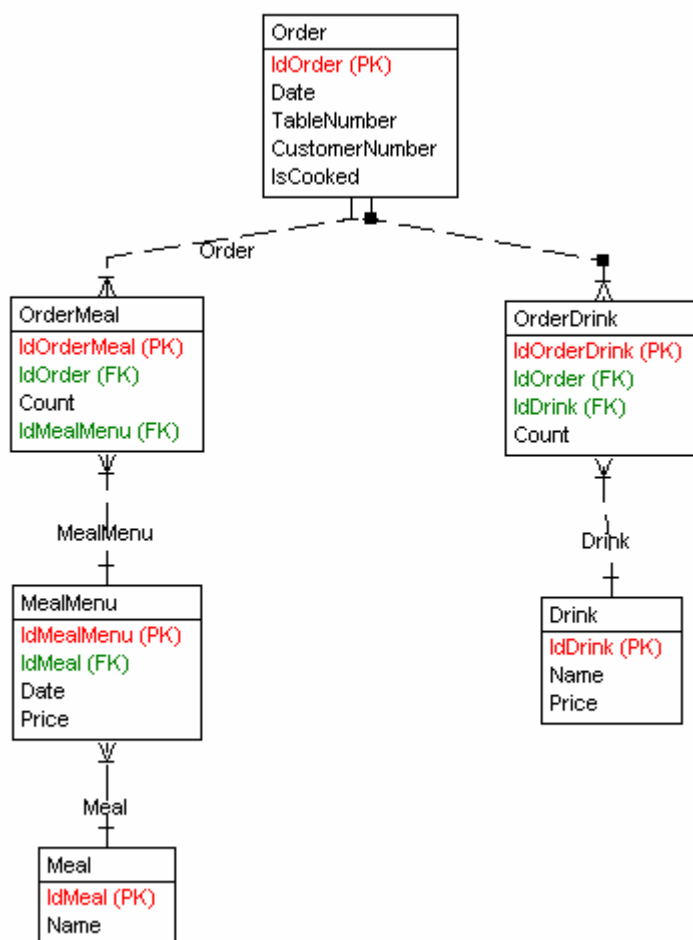
Obr. 10: Ukázka aplikace: Nastavení filtrů objednávek

Manažer restauračního zařízení má také možnost exportovat potřebná data do souboru Excel.

List 1				
	A	B	C	D
1				
2			Přehled zpotřeby za měsíc	
3				
4			Prosinec , 2006	
5				
6	Jídlo	Ks	Cena	
7				
8	Kuře na kari, rýže	1	70	
9	Kuřecí řízek, hranolky	1	55	
10	Smažený sýr, brambor	1	65	
11		Celkem	190	

Obr. 11: Ukázka aplikace: Export přehledu objednávek do souboru Excel

3.2.2 Propojení dat v systému



Obr. 12: Databázová strukturu IS RestSoft

3.2.3 Výstupy

Standardními výstupy jsou:

- tiskové sestavy všech významných dat
- export dat do souborů aplikace MS Excel

3.2.4 Diagramy tříd

Pro návrh struktury aplikace byl použit návrhový vzor Bridge(most).

Struktura aplikace se bude skládat ze 3 vrstev: uživatelské vrstvy, business logiky a datové vrstvy.

Datová vrstva

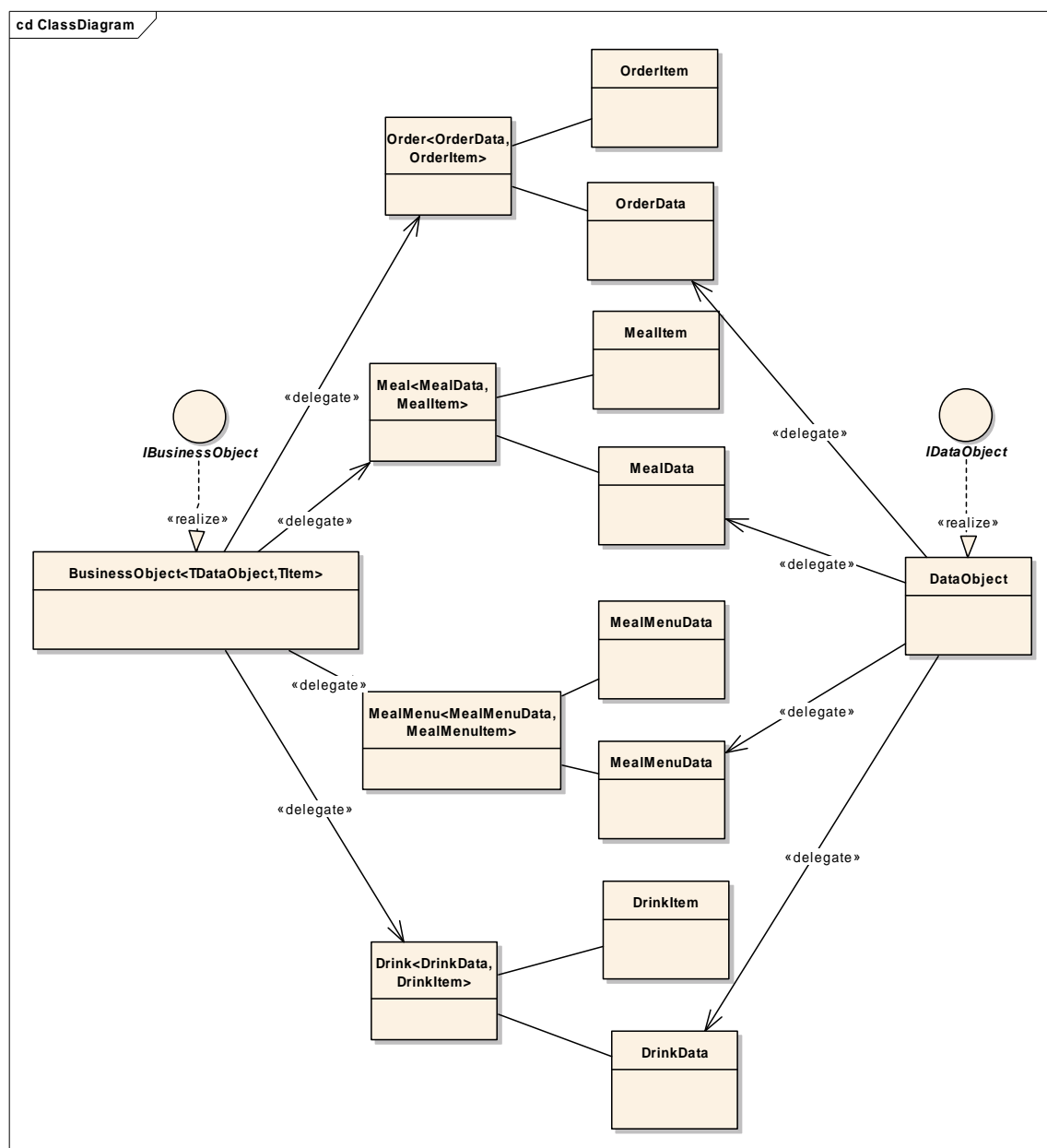
Datová vrstva je nejnižší ve struktuře. Tato vrstva zapouzdřuje logiku práce s databází. S datovou vrstvou pracuje business logika systému, žádá jí o data nebo jí data předává k uložení do DB a datová vrstva data vrací zpátky business vrstvě.

Datová vrstva je uložena v namespace Rest.Data

Business logika

Business vrstva je prostřední vrstva systému. Pracuje jak s uživatelskou vrstvou, tak s datovou vrstvou. Business vrstva vždy pouze zpracovává požadavky uživatelské vrstvy a předává je vrstvě datové. Uživatelské vrstvě poskytuje různé nástroje a prostředky (speciální třídy zapouzdřující práci s daty systému) pro uchování a předání dat datové vrstvě. Tím pádem business vrstva slouží jako prostředník (most) mezi uživatelskou a business vrstvou.

Business vrstva je uložena v namespace Rest.Business



Obr. 13: Class diagram business a datové vrstvy

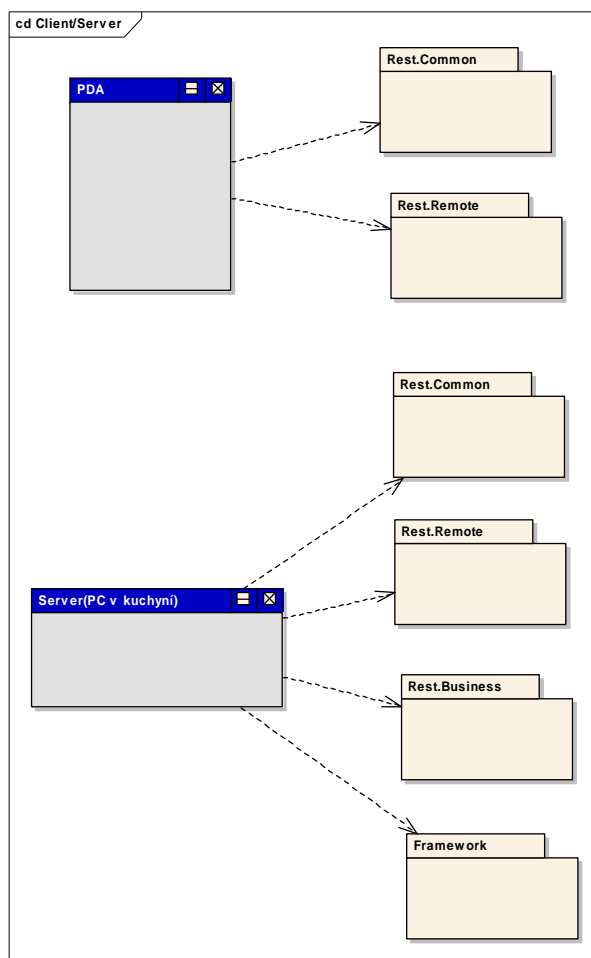
Uživatelská vrstva

Uživatelská vrstva leží ve struktuře nejvýše. Zapouzdřuje všechnu vizuální logiku systému. Používá prostředky business vrstvy k předávání dat datové vrstvě.

Tato vrstva je klientská. Jedna část uživatelské vrstvy „poběží“ na PDA (mobilní aplikace), druhá – na počítači v kuchyni – WinForm aplikace.

Protože tento informační systém pracuje na zařízení PDA, je potřeba přemýšlet o rozložení knihoven (.dll) v souborové struktuře. PDA zařízení nemusí mít dost paměti pro uchování všech knihoven. Proto celá business a datová logika je uložena na straně serveru. PDA má k dispozici namespace Rest.Remote, kde jsou uloženy funkce pro propojení se serverem. V knihovně Rest.Remote je zapouzdřena logika práce s webovými službami.

Dále máme namespace `Rest.Common`, kde jsou některé společné funkce pro klienta a server.



Obr. 14: Použití knihoven systému na straně klienta a serveru.

4 Požadavky

4.1 Systémové požadavky

Vývoj aplikace bude probíhat v prostředí Microsoft Visual Studio 2005. Běžové prostředí vyvinutých aplikací bude Microsoft .NET Framework v2.0.

4.2 Výkonové požadavky

Při splnění specifikovaných systémových požadavků jsou proměnné výkonové parametry systému plně odvislé od kvality připojení k internetu. Jiné závislosti mající zásadní vliv na celkový výkon systému nebyly v době vytváření tohoto podkladu známy.

	Název dokumentu: Specifikace IS BuildingSoft pro stavbyvedoucího
--	--

Zpracoval:	Oxana Shamanina
Verze dokumentu:	
Datum vytvoření:	14.11.2006
Počet stran:	15

Revize:

Datum	Verze	Popis	Autor
14.11.2006	1.0	Vytvoření dokumentu	Oxana Shamanina

Obsah – Příloha B

Obsah – Příloha B	70
Úvod.....	71
Účel.....	71
1 Oblasti k řešení	71
1.1 Popis současného stavu.....	71
1.2 Význam vyvíjené aplikace.....	71
2 Popis okolí systému	72
2.1 Popis uživatelů.....	72
3 Charakteristika produktu	73
3.1 Koncepce řešení.....	73
3.2 Vlastnosti systému	74
3.2.1 Popis užití a ukázky z aplikace	74
3.2.2 Propojení dat v systému.....	82
3.2.3 Výstupy.....	83
4 Požadavky.....	83
4.1 Systémové požadavky	83
4.2 Výkonové požadavky	83

Úvod

Cílem tohoto dokumentu je shromáždění, analýza a definice požadavků a vlastností systému „BuildingSoft“ včetně navrhovaných variant řešení. Dokument se zaměřuje na definici a zdůvodnění potřeb zákazníka a cílových uživatelů.

Účel

Účelem tohoto dokumentu je definovat požadavky na nejvyšší úrovni abstrakce a tedy určit rozsah vyvíjené aplikace. Informace jsou podány tak, aby byly zcela srozumitelné pro zákazníka.

1. Oblasti k řešení

1.1 Popis současného stavu

V současné době zákazník (stavbyvedoucí) používá následující způsoby evidence informací:

Stavbyvedoucí zapisuje všechny informace na papír a poté je pracně přepisuje do počítače v kanceláři. Největším problémem je dvojitá práce a také ten fakt, že data na firemním serveru nejsou aktuální, stavbyvedoucí na přepisování informací z papíru většinou nemá čas a dodává je do systému na poslední chvíli na konci období.

Informace zapsané na papír obsahují spoustu chyb, jelikož takový systém nemá žádný řád. Také se často stává, že se data ztratí úplně nebo částečně.

Na druhé straně stavbyvedoucí v některých případech potřebuje přehled o aktuálních datech. Například při vedení stavové evidence potřebuje mít přehled o aktuálním stavu materiálu na skladu (stavbě).

1.2 Význam vyvíjené aplikace

Hlavní přínos aplikace „BuildingSoft“ spočívá v usnadnění a zpřehlednění práce stavbyvedoucího, zrychlení předávání informací mezi stavbou a kanceláří, zmenšení papírování, automatizace práce. Mezi očekávanými pozitivními důsledky by mělo být:

- evidence docházky pracovníků na stavbě a následný výpočet mezd v účtárně.
- stavební evidence – evidence stavu materiálu na skladě (stavbě)
- evidence čerpání položek rozpočtu
- synchronizace dat mezi klientem a serverem

2 Popis okolí systému

Tato kapitola popisuje okolí a hranice systému – to znamená uživatele a jejich rozdělení podle jejich vztahu k vyvíjenému informačnímu systému. Dále popisuje strukturu a význam uživatelských rolí k jednotlivým entitám aplikace.

2.1 Popis uživatelů

Tato kapitola popisuje rozdělení uživatelů, definuje jejich role a typy podle jejich vztahu k informačnímu systému. Cílovou skupinu uživatelů aplikace „BuildingSoft“ tvoří stavbyvedoucí na stavbě.

Pro role, které jednotliví uživatelé v systému zastávají, se v modelování užívá výraz aktéři.

Aktér	Funkce
Stavbyvedoucí	Je to uživatel, který má u sebe přístroj (mobilní zařízení PDA) nebo notebook, ukládá do systému docházku pracovníků, stavební evidenci a čerpá rozpočet stavby. Díky aktuálním informacím má přehled o stavu stavby, materiálu a docházky. Synchronizuje data ze svého přístroje s databází na serveru ve firmě, tím udržuje serverovou databázi aktuální, nebo v případě čerpání se data po připojení synchronizují sama.

Tab. 1: Role uživatelů ve vztahu k aplikaci

3 Charakteristika produktu

3.1 Koncepce řešení

Koncepce výsledného řešení vychází z následujících předpokladů:

- systém je vybudován za užití objektových postupů návrhu vícevrstvé architektury, kde je obchodní logika aplikace prováděna odděleně od prezentační části aplikace
- všechna data jsou ukládána v relační databázi
- databáze na serveru (serverový počítač v kanceláři) je vytvořena jen pro ukázkou synchronizace. IS se nezabývá funkčností serverové aplikace
- uživatelské rozhraní na klientu (mobilní zařízení) je vytvořeno jako mobilní aplikace (tlustý klient)
- mobilní databáze na klientu je vytvořena pouze pro lokální přístup z klienta a po připojení k počítači v kanceláři bude možné synchronizovat tuto databázi.
- pro čerpání stavebního rozpočtu je použit Offline aplikační blok, který zajišťuje funkčnost aplikace v režimu offline.

Návrh architektury

Hlavní princip návrhu je založen na propojení klienta (PDA nebo notebooku) se serverem.

Pro evidenci docházky zaměstnanců na stavbu a stavební evidenci není důležitá okamžitá aktualizace dat na serveru (kancelář firmy). Proto postačí, když data z těchto součástí aplikace uložíme do lokální mobilní databáze a stavbyvedoucí jednou za čas data synchronizuje se serverovou databází.

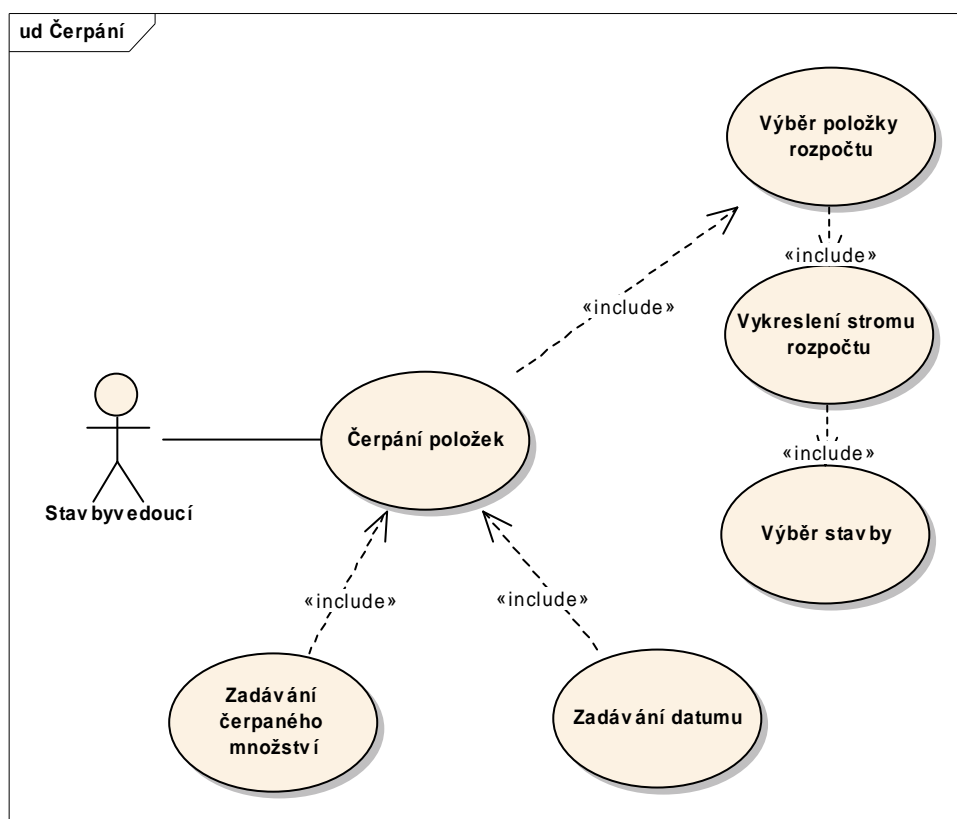
Problém je, že čerpání stavebního rozpočtu ovšem nejenom že pracuje s větším objemem dat, ale také vyžaduje častější aktualizaci dat se serverem. Proto je tento problém vyřešen pomocí webových služeb nebo remotingu, ale stavbyvedoucí nemusí být vždy připojen k síti. Aplikace sama zjistí stav připojení a pracuje s daty buď z cache v případě offline režimu nebo přímo ze serverové databáze.

Business vrstva a datová vrstva budou rozděleny na dvě části, jedna bude umístěna na serveru a druhá na klientovi.

3.2 Vlastnosti systému

3.2.1 Popis užití a ukázky z aplikace

Čerpání stavebního rozpočtu

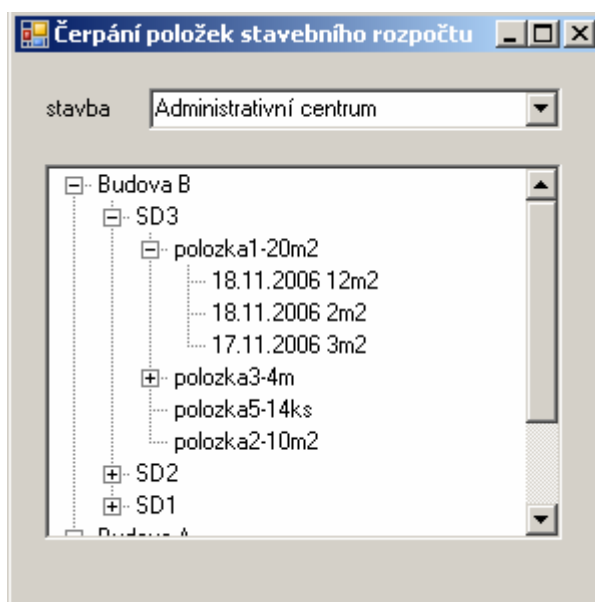


Obr. 1: Use case diagram: čerpání stavebního rozpočtu

Čerpání položek stavebního rozpočtu bude prováděno v režimu offline i online současně. To znamená, že stavbyvedoucí nemusí být vždy připojen k síti. Aplikace sama zjistí stav připojení a pracuje s daty buď z cache v případě offline režimu nebo přímo ze serverové databáze (online režim).

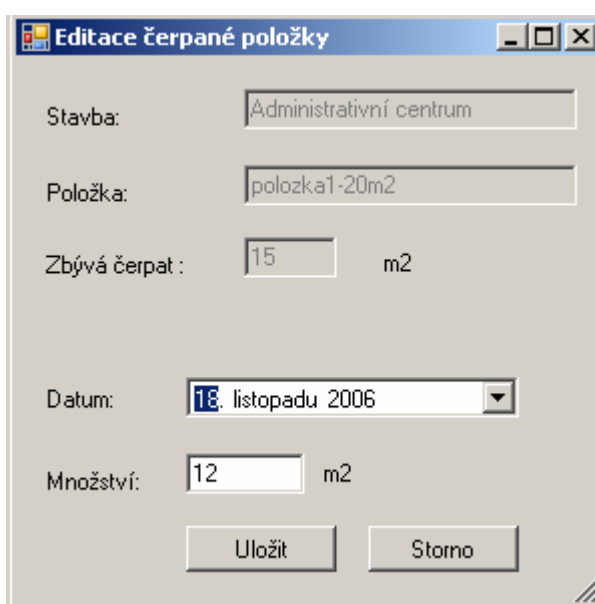
Přenos dat může být proveden buď pomocí webových služeb nebo pomocí remotingu (pokud stavbyvedoucí používá pro práci notebook).

Tato aplikace využívá funkčnosti offline aplikačních bloků, které umožňují řídit offline a online režim současně. Pokud stavbyvedoucí je připojen k síti, aplikace to pozná a synchronizuje data se serverovou databází pomocí webových služeb. Nová data rovnou odesílá na server. Když se klient od sítě odpojí, aplikace to opět pozná a pracuje s cache do té doby, než se klient znovu připojí.



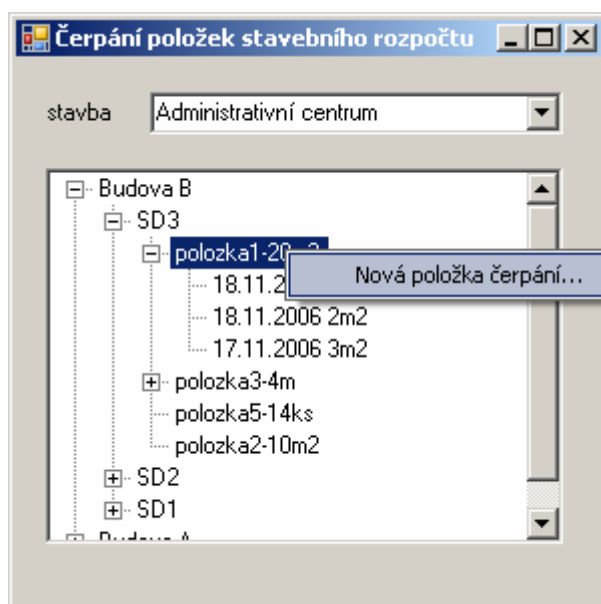
Obr. 2: Stavební rozpočet

Kliknutím na vybranou položku uživatel může čerpat položku rozpočtu dál.



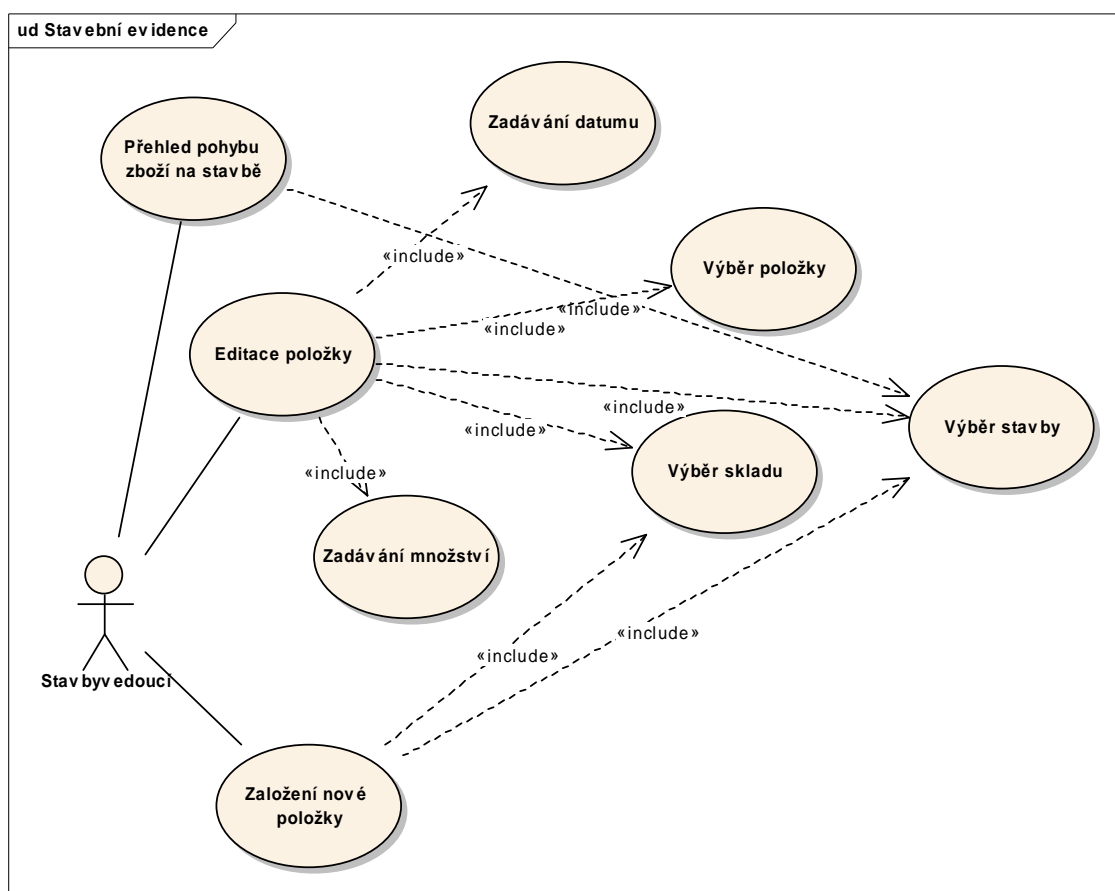
Obr. 3: Editace čerpaného množství položky rozpočtu

Stavbyvedoucí může založit novou položku čerpání nebo nějakou smazat.



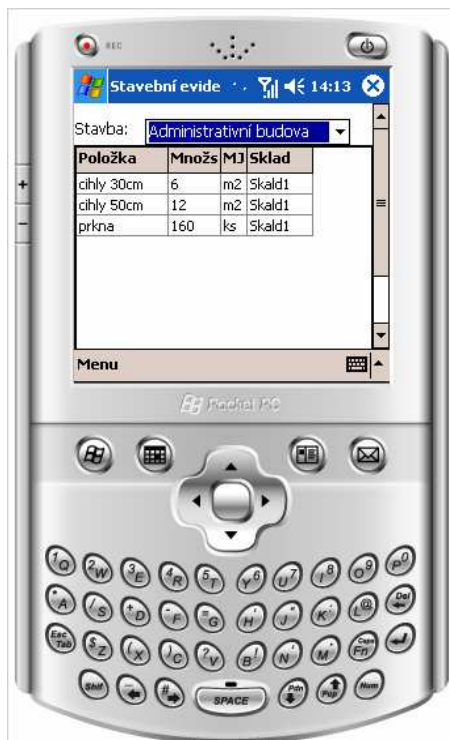
Obr. 4: Přidání nové stavební položky do rozpočtu

Stavební evidence



Obr. 5: Use case diagram: stavební evidence

Po otevření aplikace se otevře přehled všech stavebních položek na skladech pro danou stavbu a vypočítané zbývající množství.



Obr. 6: Ukázka aplikace: Přehled položek

V aplikaci je také možnost přidat novou položku a zároveň zadat množství na skladě.



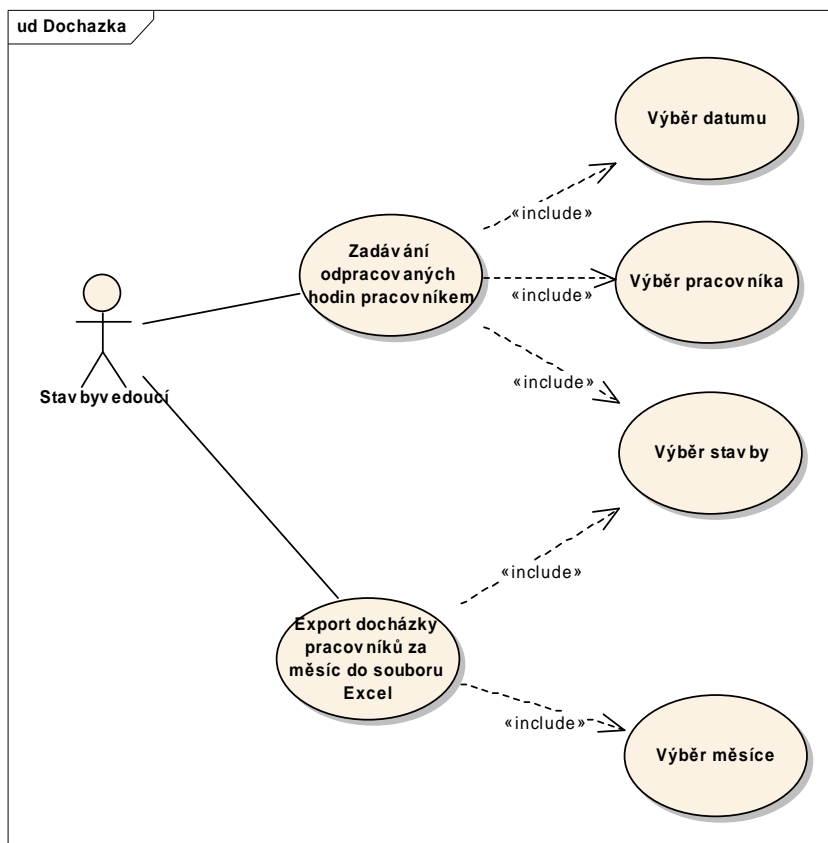
Obr. 7: Ukázka aplikace: Přidávání položky na sklad

Při kliknutí na položku se zobrazí přehled čerpání položky na stavbě a stavbyvedoucí může přidat nebo čerpat množství dané položky.



Obr. 8: Ukázka aplikace: Čerpání položky

Docházka zaměstnanců na stavbě



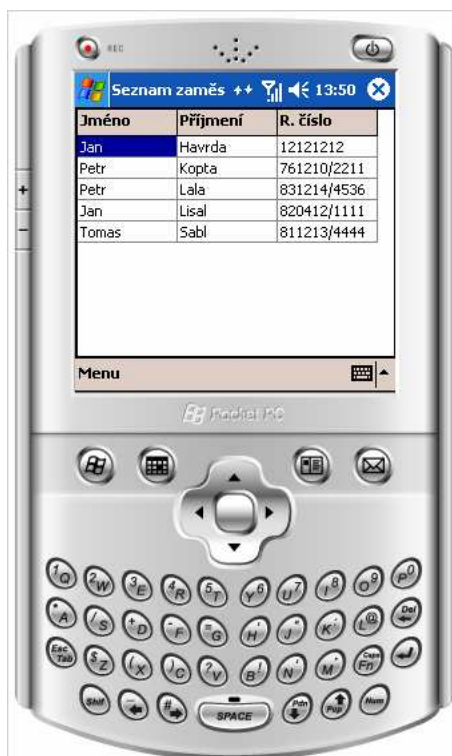
Obr. 9: Use case diagram: Docházka zaměstnanců

Jednou z funkcí aplikace je zadávání nového zaměstnance.



Obr. 10: Ukázka aplikace: Zadávání nového zaměstnance

Po otevření aplikace uživatel uvidí seznam všech zaměstnanců.



Obr. 11: Ukázka aplikace: Přehled všech zaměstnanců

Pokud vybere jednoho ze zaměstnanců, může mu zadávat docházku.



Obr. 12: Ukázka aplikace: Zadávání docházky

Stavbyvedoucí si také může nechat zobrazit docházku zaměstnance za měsíc



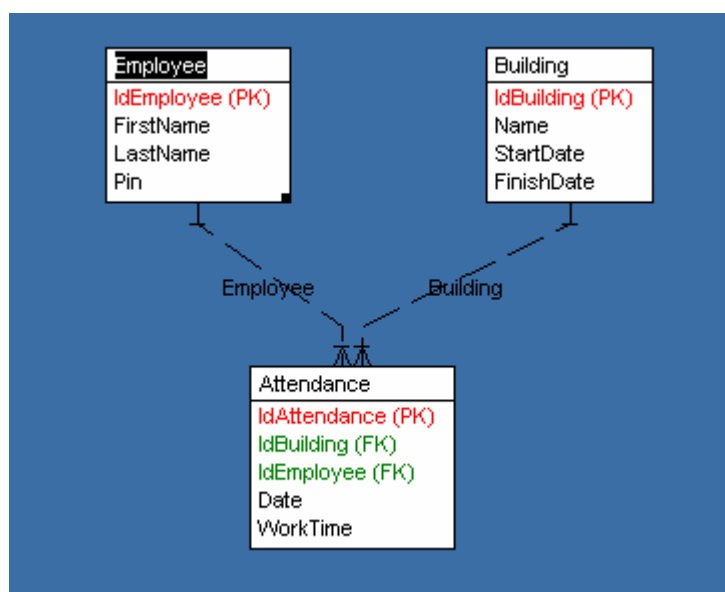
Obr. 13: Ukázka aplikace: Docházka zaměstnance za měsíc

Také je důležité mít přehled o celkové docházce zaměstnanců za měsíc

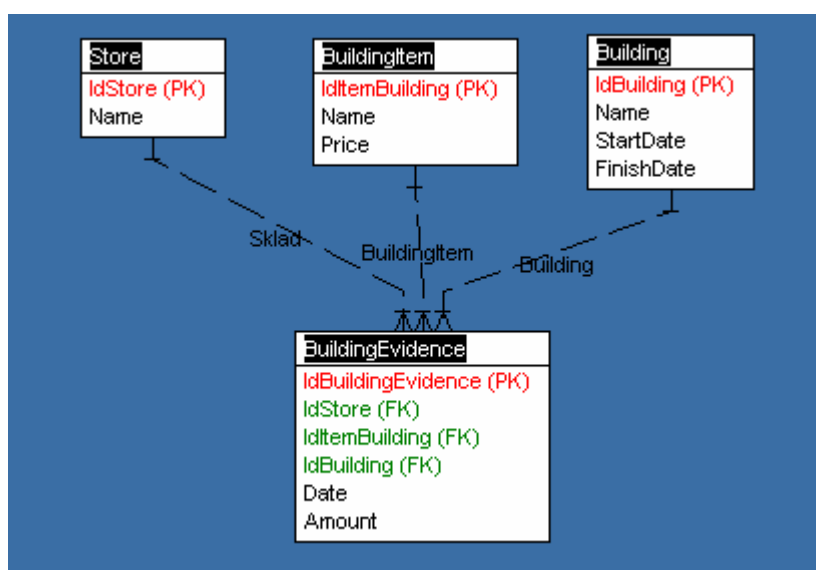


Obr. 14: Ukázka aplikace: Docházka zaměstnanců za měsíc

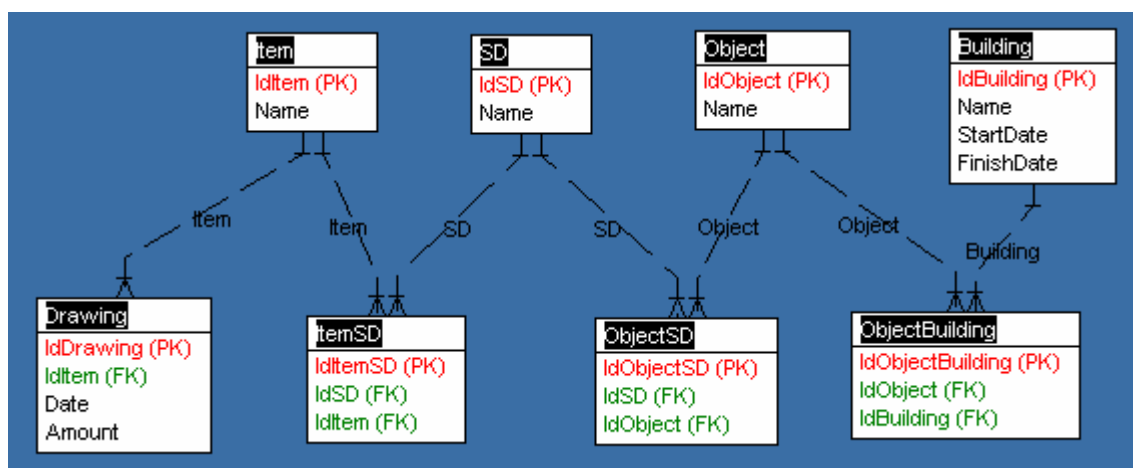
3.2.2 Propojení dat v systému



Obr. 15: Struktura databáze pro zadávání docházky



Obr. 16: Struktura databáze pro vedení stavební evidence



Obr. 17: Struktura databáze pro čerpání položek rozpočtu

3.2.3 Výstupy

Standardními výstupy jsou:

- tiskové sestavy všech významných dat
- export dat do souborů aplikace MS Excel

4 Požadavky

4.1 Systémové požadavky

Vývoj aplikace bude probíhat v prostředí Microsoft Visual Studio 2005. Běžové prostředí vyvinutých aplikací bude Microsoft .NET Framework v2.0.

4.2 Výkonové požadavky

Při splnění specifikovaných systémových požadavků jsou proměnné výkonové parametry systému plně odvislé od kvality WAN. Jiné závislosti mající zásadní vliv na celkový výkon systému nebyly v době vytváření tohoto podkladu známy.